

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-312453

(43)Date of publication of application : 09.11.2001

(51)Int.Cl.

G06F 13/10

G06F 9/06

(21)Application number : 2000-128701

(71)Applicant : CANON INC

(22)Date of filing : 28.04.2000

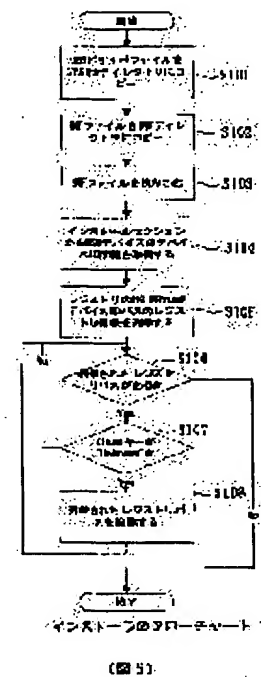
(72)Inventor : YAMADA KOJI

(54) ELECTRONIC EQUIPMENT, CONTROL METHOD FOR ELECTRONIC EQUIPMENT, AND COMPUTER-READABLE STORAGE MEDIUM WITH CONTROL PROGRAM FOR ELECTRONIC EQUIPMENT STORED THEREIN

(57)Abstract:

PROBLEM TO BE SOLVED: To make it easy to handle a peripheral equipment device connected to an interface where plural devices can be connected and its device driver.

SOLUTION: A USB driver file is copied to a SYSTEM directory (S101) and an INF file is copied to an INF directory (S102). The INF file is read in a memory (S103) and device ID information of USB in the INF file is obtained (S104). A registry of the system is read in (S105) so that it can be edited, a registry path is generated by using the device ID obtained in the step S104, and registry information below the path is listed; when a Class key is 'Unknown' (S107), information on the registry path is all deleted. When not 'Unknown', a return to a step S106 is made so as to check a next listed registry path.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2001-312453
(P2001-312453A)

(43) 公開日 平成13年11月9日 (2001.11.9)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード* (参考)
G 0 6 F 13/10	3 2 0	G 0 6 F 13/10	3 2 0 A 5 B 0 1 4
9/06	4 1 0	9/06	- 4 1 0 C 5 B 0 7 6

審査請求 未請求 請求項の数15 O L (全 34 頁)

(21) 出願番号 特願2000-128701 (P2000-128701)

(22) 出願日 平成12年4月28日 (2000.4.28)

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 発明者 山田 浩示

東京都大田区下丸子3丁目30番2号 キヤ
ノン株式会社内

(74) 代理人 100075292

弁理士 加藤 卓

Fターム(参考) 5B014 EB01 FA11 FB04

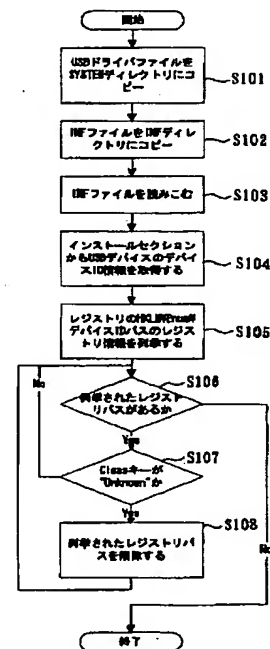
5B076 AA04 AA06 AA20

(54) 【発明の名称】 電子機器、電子機器の制御方法、および電子機器の制御プログラムを格納したコンピュータ読取可能な記憶媒体

(57) 【要約】

【課題】 複数機器を接続可能なインタフェースに接続される周辺機器デバイスないしそのデバイスドライバの取り扱いを容易にする。

【解決手段】 USBドライバファイルをSYSTEMディレクトリにコピーし (S101)、INFファイルをINFディレクトリにコピーする (S102)。INFファイルをメモリに読み込み (S103)、INFファイル内のUSBのデバイスID情報を取得する (S104)。システムのレジストリを編集できるように読み込み (S105)、ステップS104で取得したデバイスIDを用いてレジストリパスを作り、このパス以下のレジストリ情報を列挙し、Classキーが“Unknown”になっている場合 (S107) はそのレジストリパスの情報を全て削除する。“Unknown”でないなら次の列挙されたレジストリパスを調べるためステップS106へ戻る。



インストールのフローチャート

(図5)

【特許請求の範囲】

【請求項 1】 周辺機器デバイスを当該機器のデバイスドライバ経由で制御することにより当該機器に対する入出力を行なうとともに、デバイスドライバの登録に関するレジストリ情報がレジストリ情報記憶手段に登録される電子機器において、

所定の周辺機器のデバイスドライバをシステムにインストールする際、登録すべきデバイスドライバに対応したデバイス ID 情報を前記レジストリ情報記憶手段で検索し、検索されたレジストリ情報の一部が不明となっていた場合、当該レジストリ情報を削除することを特徴とする電子機器。

【請求項 2】 所定の周辺機器のデバイスドライバをシステムにインストールする際、前記デバイス ID 情報がインストール情報定義ファイルから取り出されることを特徴とする請求項 1 に記載の電子機器。

【請求項 3】 複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続し、周辺機器デバイスを当該機器のデバイスドライバ経由で制御することにより当該機器に対する入出力を行なう電子機器において、

前記入出力ポートが増加したかどうか判断し、前記入出力ポートが増加した場合に増加したポートを介してそのポートに対応する周辺機器のデバイス名を取得し、取得したデバイス名が特定の対応機種を示している場合に当該周辺機器のデバイスドライバのインストール処理を行なうことを特徴とする電子機器。

【請求項 4】 複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続して当該機器に対する入出力を行なう電子機器において、ユーザが入出力先として選択すべき周辺機器を確認するための確認コマンドを前記入出力ポート経由で送信し、該確認コマンドを受信した周辺機器側において該確認コマンドの受信をユーザに報知することを特徴とする電子機器。

【請求項 5】 複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続して当該機器に対する入出力を行なう電子機器において、前記周辺機器の操作または動作状態に応じて定まるステータス情報を前記入出力ポートを介して前記周辺機器デバイスから受信し、受信したステータス情報に応じて入出力先として周辺機器を選択することを特徴とする電子機器。

【請求項 6】 周辺機器デバイスを当該機器のデバイスドライバ経由で制御することにより当該機器に対する入出力を行なうとともに、デバイスドライバの登録に関するレジストリ情報がレジストリ情報記憶手段に登録される電子機器の制御方法において、

所定の周辺機器のデバイスドライバをシステムにインストールする際、登録すべきデバイスドライバに対応した

デバイス ID 情報を前記レジストリ情報記憶手段で検索し、検索されたレジストリ情報の一部が不明となっていた場合、当該レジストリ情報を削除することを特徴とする電子機器の制御方法。

【請求項 7】 所定の周辺機器のデバイスドライバをシステムにインストールする際、前記デバイス ID 情報がインストール情報定義ファイルから取り出されることを特徴とする請求項 6 に記載の電子機器の制御方法。

【請求項 8】 複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続し、周辺機器デバイスを当該機器のデバイスドライバ経由で制御することにより当該機器に対する入出力を行なう電子機器の制御方法において、

前記入出力ポートが増加したかどうか判断し、前記入出力ポートが増加した場合に増加したポートを介してそのポートに対応する周辺機器のデバイス名を取得し、取得したデバイス名が特定の対応機種を示している場合に当該周辺機器のデバイスドライバのインストール処理を行なうことを特徴とする電子機器の制御方法。

【請求項 9】 複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続して当該機器に対する入出力を行なう電子機器の制御方法において、ユーザが入出力先として選択すべき周辺機器を確認するための確認コマンドを前記入出力ポート経由で送信し、該確認コマンドを受信した周辺機器側において該確認コマンドの受信をユーザに報知することを特徴とする電子機器の制御方法。

【請求項 10】 複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続して当該機器に対する入出力を行なう電子機器の制御方法において、前記周辺機器の操作または動作状態に応じて定まるステータス情報を前記入出力ポートを介して前記周辺機器デバイスから受信し、受信したステータス情報に応じて入出力先として周辺機器を選択することを特徴とする電子機器の制御方法。

【請求項 11】 周辺機器デバイスを当該機器のデバイスドライバ経由で制御することにより当該機器に対する入出力を行なうとともに、デバイスドライバの登録に関するレジストリ情報がレジストリ情報記憶手段に登録される電子機器の制御プログラムを格納したコンピュータ読取可能な記憶媒体において、所定の周辺機器のデバイスドライバをシステムにインストールする際、登録すべきデバイスドライバに対応したデバイス ID 情報を前記レジストリ情報記憶手段で検索し、検索されたレジストリ情報の一部が不明となっていた場合、当該レジストリ情報を削除するための制御手順を格納したことを特徴とする電子機器の制御プログラムを格納したコンピュータ読取可能な記憶媒体。

【請求項 12】 所定の周辺機器のデバイスドライバをシステムにインストールする際、前記デバイス ID 情報

がインストール情報定義ファイルから取り出されることを特徴とする請求項11に記載の電子機器の制御プログラムを格納したコンピュータ読取可能な記憶媒体。

【請求項13】 複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続し、周辺機器デバイスを当該機器のデバイスドライバ経由で制御することにより当該機器に対する入出力を行なう電子機器の制御プログラムを格納したコンピュータ読取可能な記憶媒体において、

前記入出力ポートが増加したかどうか判断し、前記入出力ポートが増加した場合に増加したポートを介してそのポートに対応する周辺機器のデバイス名を取得し、取得したデバイス名が特定の対応機種を示している場合に当該周辺機器のデバイスドライバのインストール処理を行なうための制御手順を格納したことを特徴とする電子機器の制御プログラムを格納したコンピュータ読取可能な記憶媒体。

【請求項14】 複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続して当該機器に対する入出力を行なう電子機器の制御プログラムを格納したコンピュータ読取可能な記憶媒体において、ユーザが入出力先として選択すべき周辺機器を確認するための確認コマンドを前記入出力ポート経由で送信し、該確認コマンドを受信した周辺機器側において該確認コマンドの受信をユーザに報知するための制御手順を格納したことを特徴とする電子機器の制御プログラムを格納したコンピュータ読取可能な記憶媒体。

【請求項15】 複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続して当該機器に対する入出力を行なう電子機器の制御プログラムを格納したコンピュータ読取可能な記憶媒体において、前記周辺機器の操作または動作状態に応じて定まるステータス情報を前記入出力ポートを介して前記周辺機器デバイスから受信し、受信したステータス情報に応じて入出力先として周辺機器を選択するための制御手順を格納したことを特徴とする電子機器の制御プログラムを格納したコンピュータ読取可能な記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続する電子機器、その制御方法、およびその制御プログラムを格納したコンピュータ読取可能な記憶媒体に関するものである。

【0002】

【従来の技術】 従来より、プリンタなどの周辺機器がコンピュータシステムとともに広く利用されている。従来では、プリンタはセントロニクスポートやRS232Cのようなシリアルポートを介して、1ポートに1台のみのプリンタをコンピュータ本体に接続することが多かつ

たが、近年になっていわゆるUSBポートのような、より汎用的なインタフェースを介して1ポートに複数台のプリンタを接続する形態が増えつつある。

【0003】

【発明が解決しようとする課題】 一方、マイクロソフト社のWindows（商品名：Windows95、98、2000、NTなどの異なるバージョンがある）ファミリーがパーソナルコンピュータのOSとして広く用いられているが、USBデバイスのサポートを標榜しているものの、現在のところ標準で添付しているデバイスドライバはあまり多くない。

【0004】 たとえば、Windows98におけるUSBドライバのインストール手順は、他のドライバのインストールとほぼ同様、いわゆるSETUP.EXEのようなインストーラプログラムを手動で、あるいは「ウィザード」と呼ばれるプログラムにより、OSの起動時や、デバイスのPnP（プラグ・アンド・プレイ）操作により自動ないし半自動的に起動することにより行なわれる。

【0005】 たとえば、OS起動後にUSBデバイスをパソコンに接続すると、OSはUSBデバイスを検知し、ウィザードプログラムのデバイスドライバインストールダイアログを表示させる。このダイアログで、ユーザはデバイスドライバが格納されているディレクトリを正しく指定する必要がある。

【0006】 ユーザがダイアログを介してデバイスドライバが格納されているディレクトリ（CD-ROMやフロッピー（登録商標）ディスク上のディレクトリ）を指定すると、OSは指定されたディレクトリに検知したUSBデバイス用のデバイスドライバが存在するか確認し、対応したものが存在すれば、これをインストールして終了する。

【0007】 ところが、ユーザが間違ったディレクトリを指定したため、検知したUSBデバイスに対応していないデバイスドライバがあったり、そもそも当該のディレクトリにデバイスドライバが存在しない場合、OSは検知したUSBデバイスを「不明なデバイス」として認識し、以後、そのデバイスは正常に機能しなくなる。この状態では、レジストリと呼ばれるOSの設定データに誤まったデータが登録されたままとなっており、このままでは再度USBデバイスを繋ぎなおしても「不明なデバイス」として認識するため、ユーザはデバイスを全く使用できなくなる。

【0008】 このような状態になってしまった場合は、ユーザはデバイスマネージャなどのプログラムを用いて手動により当該デバイスを明示的に削除して正しい操作をやり直さなければ正常なデバイスドライバのインストールを行なうことができない。また、このような措置はある程度の経験を積んだユーザでなければ行なうのは困難であり、初心者では途方に暮れてしまう。また、この種

の些細なトラブルに関するユーザサポートに法外に高額な料金を要求するベンダーもあり、初心者であっても確実に失敗なくUSBデバイス（ドライバ）のインストールを行なえるようにすることが望まれているのはいうまでもない。

【0009】また、プリンタなどの機器をUSBポートに接続して用いる場合、この種のポートには複数の機器を同時に接続して用いることができるため、従来、存在しなかった新たな問題が発生しつつある。

【0010】すなわち、従来のプリンタはセントロニクスインタフェースなどのホストとデバイスの一対一通信しかできないインタフェースで接続されていたため、プリンタドライバ用のインストーラは例えばセントロニクスインタフェースからプリンタと直接通信を行い、プリンタ名などの情報を得て、適切なプリンタドライバをインストールしていた。

【0011】たとえば、Windowsの場合は通信インタフェースをポートと表現し、セントロニクスインタフェースのWindows上の名称（ポート名）はLPT1である。そこでデバイス（ドライバ）のインストーラはLPT1というポートをオープンし、このポートからプリンタに対してプリンタ名を要求し、取得したプリンタ名がインストーラの対応するプリンタ名と判断できたらインストール処理を行うようになっていた。

【0012】しかし近年普及しつつあるUSBインタフェースやIEEE1394インタフェースは一つのインタフェースに複数のデバイスが接続でき、しかもアドレスが動的に割り振られるため、あらかじめポート名を決めておくことができない。そのためあらかじめ決められたポート名を調べる従来の方法ではインストーラによる適切なプリンタドライバのインストールが行えず、ユーザに負担を強いることになっていた。

【0013】より具体的に問題点を挙げると次のようになる。Microsoft社のWindowsにおけるUSBインタフェースは、ホストとデバイスを接続しない限りポートと認識されない。しかもUSBインタフェースは最大127台のデバイスを接続可能であり、ポートと認識されたとしてもどのような名称になるか予め予測することができない。そのため従来のようにあらかじめ決められたポート名を使って接続機器、たとえばプリンタ（の機種）を検出することが困難になり、ユーザが適切なドライバを選択しなければならない、という問題があった。

【0014】あるいは、Windowsにおけるプラグアンドプレイ機能を用いることで、自動的にプリンタドライバをインストールすることもできるが、ベンダー独自のアプリケーションなどをインストールすることができないため独自の機能を追加することができない。そのためプリンタドライバのインストールにはインストーラが必要不可欠である。

【0015】以上の問題は、USBポート接続のプリンタに特別なものではなく、あらかじめデバイス名やアドレスなどを予め予測できない、という意味ではイーサネット（登録商標）などのネットワークを介して接続される機器においても同様にあてはまるものである。

【0016】また、外部機器がイーサネットなどのネットワークを介して接続される構成においては、同一機能の機器（たとえばプリンタ）が複数そのポートやネットワークに同時接続されることも稀ではなく、たとえば同一機種のプリンタが複数台そのポートやネットワークに接続されることもあり得る。

【0017】製品名とは別にニックネームのような名前を付けることでプリンタを区別する方法が一般的であった。しかしこの方法はニックネームを忘れると、どのプリンタが本当に印刷したいプリンタかわからなくなっていた。そのため試しに印刷してそれがどのプリンタから出てくるか確認しなければならなかった。そこで試し印刷をすることなく印刷したいプリンタを区別する方法が望まれている。

【0018】本発明の課題は、上記の問題を解決し、USBポートなどの複数機器を接続可能なインタフェースに接続されるデバイスドライバを誰でも間違いなくシステムにインストールでき、USBポートなどの複数機器を接続可能なインタフェースに接続されるプリンタなどの外部機器の識別を容易に行なうことができ、自分が使用したいプリンタなどの機器を容易に確認できるようにすることにある。

【0019】

【課題を解決するための手段】上記の課題を解決するため、本発明においては、周辺機器デバイスを当該機器のデバイスドライバ経由で制御することにより当該機器に対する入出力を行なうとともに、デバイスドライバの登録に関するレジストリ情報がレジストリ情報記憶手段に登録される電子機器、その制御方法、およびその制御プログラムを格納したコンピュータ読取可能な記憶媒体において、所定の周辺機器のデバイスドライバをシステムにインストールする際、登録すべきデバイスドライバに対応したデバイスID情報を前記レジストリ情報記憶手段で検索し、検索されたレジストリ情報の一部が不明となっていた場合、当該レジストリ情報を削除する構成を採用した。

【0020】あるいはさらに、所定の周辺機器のデバイスドライバをシステムにインストールする際、前記デバイスID情報がインストール情報定義ファイルから取り出される構成を採用した。

【0021】あるいは、複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続し、周辺機器デバイスを当該機器のデバイスドライバ経由で制御することにより当該機器に対する入出力を行なう電子機器、その制御方法、およびその制御プログラムを格納

したコンピュータ読取可能な記憶媒体において、前記入出力ポートが増加したかどうか判断し、前記入出力ポートが増加した場合に増加したポートを介してそのポートに対応する周辺機器のデバイス名を取得し、取得したデバイス名が特定の対応機種を示している場合に当該周辺機器のデバイスドライバのインストール処理を行なう構成を採用した。

【0022】あるいは、複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続して当該機器に対する入出力を行なう電子機器、その制御方法、およびその制御プログラムを格納したコンピュータ読取可能な記憶媒体において、ユーザが入出力先として選択すべき周辺機器を確認するための確認コマンドを前記入出力ポート経由で送信し、該確認コマンドを受信した周辺機器側において該確認コマンドの受信をユーザに報知する構成を採用した。

【0023】あるいは、複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続して当該機器に対する入出力を行なう電子機器、その制御方法、およびその制御プログラムを格納したコンピュータ読取可能な記憶媒体において、前記周辺機器の操作または動作状態に応じて定まるステータス情報を前記入出力ポートを介して前記周辺機器デバイスから受信し、受信したステータス情報に応じて入出力先として周辺機器を選択する構成を採用した。

【0024】

【発明の実施の形態】以下、図を参照して本発明の実施の形態を説明する。

【0025】〔第1の実施形態〕本実施形態では、Windows（商品名）上のデバイスドライバをインストールする処理を例示する。本実施形態では、デバイスドライバのインストールの際、いわゆるINFを参照してセットアップを実行するが、システムのレジストリを検索して対応するデバイスが「不明」なデバイスとなっている場合に自動的にレジストリから該当するレジストリパスを削除することにより、明示的な削除操作などを必要とすることなくデバイスドライバを誰でも間違いなくシステムにインストールできるようにする。

【0026】図1に本実施形態を実現するためのハードウェア構成を説明する。図示の構成は公知のパーソナルコンピュータの構成としてごく一般的なものである。

【0027】図1において符号H01は本実施形態の制御を司る制御部（以後CPUと称する）、H02は各種データを入力するキーボード、H03はRAMH04に貯えられている画像データを表示するカラー表示器で、CRTあるいはLCDディスプレイなどから構成される。

【0028】符号H04は装置全体を制御する制御手順およびその他の必要な情報をあらかじめ記憶するリードオンリーメモリ（以下ROMという）、H05はワーク

エリアとして利用されるランダムアクセスメモリ（以下RAMという）、H06は各種データの読み書きやプログラムが格納されているハードディスクである。後述のセットアッププログラムは、ハードディスクH06や不図示のCD-ROM、あるいはフロッピーディスクなどに格納された状態で供給され、RAMH05上（あるいはさらにハードディスクH06上に実装された仮想記憶機構を利用して）で実行される。

【0029】符号H07はUSBインタフェースであり、H08のUSBデバイスが接続される。USBデバイスH08としてはプリンタ、キーボード、マウス、タブレット、サウンドデバイス、デジタルカメラなど任意のデバイスが考えられる。H09はデータバスであり、アドレスバス、データバス、各種制御信号線などから構成され、各種データを転送するために用いられる。

【0030】図2に、本実施形態でインストールされるデバイスドライバのファイル構成を示す。図2のUSBDEV.SYSはUSBデバイス（この場合プリンタ）ドライバのファイルであり、動的にシステムに組み込むことができる一種の実行ファイルの形態で記述されたものである。

【0031】また、USBDEV.INFは上記のドライバをインストールするときに使われる設定ファイルである。その拡張子から、通称INFファイルと呼ばれている。SETUP.EXEは本発明を実施したインストールプログラムである。図2の各ファイルはハードディスクH06や不図示のCD-ROM、あるいはフロッピーディスクなどに格納された状態で供給される。たとえば、USB接続のプリンタを供給する場合、CD-ROM、あるいはフロッピーディスクなどに格納してそのプリンタとともにこれらのファイルが供給されるものとする。

【0032】なお、USBポートそのもののドライバは別途インストールされるものとする。

【0033】図3に上記のUSBDEV.INFの内容を示す。INFファイルは図示のように□で囲んだ名前によりセクションを定義し、その後に所定の制御パラメータのモニタとそのパラメータに定義すべき値を配列したテキストファイルであり、図中、左端の数字は行番号でありデータの一部をなすものではない。

【0034】1行目から4行目の[Version]セクションは、デバイスがUSBデバイスであり、デバイスのベンダー（Provider）は“Qanon”であることを示している。

【0035】続くセクション[Destination Dir]は、（5行目から6行目）はインストール先ディレクトリがWINDOWS¥SYSTEMディレクトリであることを示している。

【0036】[Manufacturer]セクション（7行目～8行目）は上記のProviderと下記の

Qanon. Sectionの関係性を定義したものである。

【0037】 [Qanon. Section] (9行目から10行目)はUSBのデバイスID "USB¥VID_04A9&PID_1052" のインストール処理を定義している。すなわち、デバイスID "USB¥VID_04A9&PID_1052" のインストールは、後述の [Qanon. USB] セクションの方法にしたがって行うことを定義している。また、ここでは名称は "Qanon USB Device" であることが定義されている。

【0038】 [Qanon. USB] セクション (11行目から13行目) においては、インストール時のファイルコピーは後述の [Qanon. CopyList] セクションの情報に従うことを定義している。また、ここではレジストリ情報の追加は後述の [Qanon. AddReg] セクションの情報に従うことを示している。

【0039】 [Qanon. CopyList] セクション (14行目から15行目) は、usbdev. sysファイル (図2) をインストールすることを示している。 [Qanon. AddReg] セクション (16行目から23行目) は、システムのレジストリに追加すべき情報である。ここでは、ドライバの名称や、デバイスの種別、デバイスIDなどが定義されている。

【0040】 次に図4を参照して、インストール時に "不明なデバイス" となった場合のレジストリの処理につき説明する。図4はレジストリエディタを用いて当該デバイスのエントリを開いた状態を示している。USBデバイスのレジストリ情報はHKEY_LOCAL_MACHINE¥Enum¥USB以下に記録される。レジストリは実際にはデータファイルの形式でハードディスクH06に記録されているが、レジストリエディタを用いてアクセスすると図示のような階層構造として表示される。「¥」はファイルシステムにおけるディレクトリのデリミタと同様に階層の境界を示す。

【0041】 本デバイスの場合、インストール操作を行なうと、図4のようにHKEY_LOCAL_MACHINE¥Enum¥USB¥VID_04A9&PID_1052以下にデバイスが持つシリアル番号キーの中にデータが記録される。

【0042】 図4はシリアル番号000000のデバイスを示しており、このレジストリ情報のClass情報がデバイスの種類を表す。図3のようなINFファイルを用いて正常にインストーラが実行された場合は、USBデバイスであればこのClass情報は "USB" となるが、インストールの際にユーザが間違ったディレクトリを指定したため、検知したUSBデバイスに対応していないデバイスドライバがあったり、そもそも当該のディレクトリにデバイスドライバが存在しない場合はこ

のClass情報の値は図4のように "不明なデバイス" を示す "Unknown" となる。

【0043】 次に、図5に本実施形態のインストールプログラムのフローチャートを示す。図示のプログラムは、図2のSETUP. EXEの一部として実装され、CD-ROM、あるいはフロッピーディスクなどに格納して供給される。

【0044】 図5のステップS101では、USBドライバファイルであるUSBDEV. SYSをWINDOWS¥SYSTEMディレクトリにコピーする。続いてステップS102ではINFファイルであるUSBDEV. INFをWINDOWS¥INFディレクトリにコピーする。

【0045】 ステップS103ではUSBDEV. INFファイルをメモリに読み込み、ステップS104ではINFファイル内の [Qanon. Section] を探し、USBのデバイスID情報を取得する。ここで、図3のINFファイルの10行目の右辺の二つ目のパラメータがUSBのデバイスIDとなる。

【0046】 ステップS105では、システムのレジストリを編集できるように読み込み、ステップS104で取得したデバイスIDの前にHKEY_LOCAL_MACHINE¥Enum¥を付加し、レジストリパスを作り、このパス以下のレジストリ情報を列挙する。図4のレジストリ内容であればデバイスID "USB¥VID_04A9&PID_1052" に相当するシリアル番号000000のデバイスが列挙される。

【0047】 ステップS106～S108のループにおいて、同一デバイスIDのレジストリパスに複数のデバイスが登録されていれば、これらは順次列挙される。

【0048】 ステップS106では列挙したレジストリパスが残っているか判断する。残っていればステップS107へ進み、残っていなければインストール処理を終了する。ステップS107ではClassキーが "Unknown" になっているか否かを判断する。 "Unknown" ならステップS108へ進み、 "Unknown" でないなら次の列挙されたレジストリパスを調べるためステップS106へ戻る。ステップS108では、Classキーが "Unknown" になっているレジストリパスの情報を全て削除する。

【0049】 このようにして、本プログラムが対象とするデバイスドライバのインストールの際にユーザが間違ったディレクトリを指定したため、検知したUSBデバイスに対応していないデバイスドライバがあったり、そもそも当該のディレクトリにデバイスドライバが存在しない場合はこのClass情報の値は図4のように "不明なデバイス" を示す "Unknown" となっているレジストリパスの情報を全て削除することができるため、以後、USBデバイスを繋ぎ直すなどの操作を行ない、正しいディレクトリを指定するなどすればその時は

プリンタなどのUSBデバイスのドライバのインストールを正常に実行することができる。

【0050】したがって、ユーザは従来のようにレジストリエディタなどを用いて明示的ドライバを削除するような操作を必要とせず、また、高額なユーザサポートを必要とせず、容易に自分自身でUSBデバイスのインストールを行なうことができる。

【0051】本発明を実施したSETUP. EXEはフロッピーディスクやCD-ROMに限定されることなく任意のインストールメディアに格納して提供することができるのはいうまでもない。

【0052】なお、上記実施形態ではUSBデバイスのデバイスIDとしてUSB¥VID_04A9&PID_1052を用いたが、それ以外のデバイスIDでも適用できることは言うまでもない。また、以上ではUSBポートを例示したが、上記の技術は複数の外部機器を接続するインタフェース、たとえばIEEE1394のようなインタフェースであっても同様に実施できるのはいうまでもない。

【0053】〔第2の実施形態〕以上では、インストーラ実行時のレジストリの操作につき示したが、以下では、USBなどのポートに動的に接続されるデバイスに適したインストーラの構成につき詳細に説明する。

【0054】本実施形態でも、Microsoft社のWindows上で動作するUSBプリンタに対応したインストーラである。USBプリンタの場合、2つのデバイスドライバを必要とする。1つはプリンタドライバ、もう1つはより低位のUSBドライバである。USBドライバはUSBでデータの送受信を行うためのデバイスドライバである。そのためUSBプリンタに対応したインストーラでは2つのデバイスドライバのインストールを行う必要がある。

【0055】本実施形態におけるインストール手順は次のようなものとなる。ユーザが本実施形態のインストーラを起動する（あるいはウィザードなどのプログラムにより自動起動される）と図7の画面が表示されるので、ユーザはUSBプリンタをポストと接続する。これにより後述のインストーラプログラムを介してUSBドライバがインストールされる。

【0056】USBポートそのもののドライバのインストール処理は前記実施形態同様に本実施形態のインストーラではなく、Windowsに付属のインストーラにより行なわれるものとする。また、USBドライバはインストールされると新しいポートを作成する。

【0057】以上の動作はWindowsにおけるUSBドライバの一般的な仕様である。一度USBドライバがインストールされると、本実施形態のインストーラは新しいポートを認識し、接続されたプリンタ名を得て最適なプリンタドライバを選択する。そしてファイルのコピー中は図8の画面を表示してファイルコピー処理を行

いプリンタドライバを登録する。ユーザはインストーラの起動とプリンタの接続だけ行えばよい。

【0058】本発明を実現するためのハードウェア構成を図6に示す。図6の構成は前記実施形態の図1とほぼ同様であるので、ここでは重複した説明は省略する。

【0059】図6において符号H07はプリンタインタフェースであり、H08はプリンタである。プリンタインタフェースH07としてはUSBが用いられるものとする。

【0060】次に図14と図15のフローチャートを用いて本実施形態のプリンタデバイスドライバのインストーラの動作を説明する。プリンタインストーラの実行ファイルの名称は前述の第1実施形態同様にSETUP. EXEであり、ハードディスクに格納された状態で、あるいは前述の実施形態同様にCD-ROMやフロッピーディスクなどのメディアにより供給される。

【0061】これらのメディア上のインストーラのバイナリ（SETUP. EXE）がユーザにより明示的に起動されるか、あるいはウィザード経由で起動されると、図14のステップS101において、USBドライバに関係するファイルをシステムにコピーする（図2参照）。USBドライバのファイルをあらかじめコピーすることで、ユーザはファイルを指定する必要がなくなる。

【0062】ステップS102はシステムが持つ全てのポート名を列挙し、ポートリストAとしてメモリに記憶するステップである。ポートリストの構造を図9に示す。図9はLPT1、COM1、FILE、およびUSBPRN01というポートを記憶した例である。

【0063】ステップS103では、図7のインストーラ画面を表示するステップで、この画面でユーザにUSBプリンタを繋ぐよう促す。ユーザがUSBポートにプリンタを繋ぐと、ステップS104においてポートチェックのインターバル時間になるまで待つ処理を入れる。本実施形態ではこのインターバル時間を2秒としている。2秒になったらステップS105へ進む。

【0064】ステップS105ではシステムが持つ全てのポート名を列挙し、ポートリストBとしてメモリに記憶する。

【0065】ステップS106ではポートリストBからポートリストAに存在するポートを削除する。この削除の結果、ポートリストBには新たに追加されたポートが残る。もしポートが追加されていないければ何も残らない。

【0066】ステップS107では、ポートリストBにポートが残っているか判断する。ポートが残っていなければステップS104へ戻り、ポートが残っていればステップS108へ進む。

【0067】ステップS108では、ポートリストBに残っているポート名がUSBインタフェースのポートか

否かを判断する。本実施形態ではUSBインタフェースのポート名は“USBPRN数字”となる。そこでポートリストBに残っているポート名の先頭が“USBPRN”かどうか判断し、先頭が“USBPRN”ならUSBインタフェースのポートだと判断する。USBインタフェースのポートだと判断したらステップS109へ進む。そうでなければステップS104へ戻る。

【0068】ステップS109ではUSBインタフェースのポートをオープンし、ステップS110ではオープンしたポートからUSBプロトコルのGET_DEWCE_IDリクエストを要求し、接続されたUSBデバイスのデバイスIDを取得する。このデバイスIDは図10に示されるような文字列である。図10のデバイスIDは、デバイスの製造者が“Qanon”で製品名は“ABC-3000”ということを示している。

【0069】図15のステップS111では、ステップS109でオープンしたUSBインタフェースのポートをクローズする。

【0070】ステップS112はデバイスIDからプラグアンドプレイID（略してPnPID）を作るステップである。このPnPIDはデバイスIDのMFGタグの名称とMDLタグの名称を足したものである。図10のデバイスIDの場合、PnPIDは“QanmABC-3000”となる。

【0071】ステップS113では、インストーラが持つ対応プリンタリストの中にステップS112で作成したPnPIDが記載されているか判断する。ここで、本実施形態の対応プリンタリストの構造を図11に示す。図11に示すように、対応プリンタリストはPnPIDと対応するファイル名から構成される。図11の例では、このインストーラは、“Qanon”社製の“ABC-300”、“ABC-80v”、“ABC-800α”の3機種に対応していることがわかる。もしステップS112で作成したPnPIDが対応プリンタリストに記載されていれば対応プリンタと判断しステップS114へ進み、記載されていなければステップS104へ戻る。

【0072】ステップS114では、ステップS103で表示したインストール画面1を消去し、ファイルのコピー中、ステップS115において図8のインストール画面2を表示する。ステップS116では、対応プリンタリストに記載されているファイル名を取得する。ステップS117では、ステップS116で取得したファイルをコピーする。

【0073】ステップS118では対応プリンタリストのファイルを全てコピーしたか判断する。全てのコピーが終了したらステップS119へ進む。コピーが終了していなければステップS117へ戻る。

【0074】ステップS119では、ステップS117でコピーしたファイルをプリンタドライバとしてシステ

ムに登録する。ステップS120ではステップS115で表示したインストール画面2を消去する。

【0075】以上のようにして、入出力ポートが増えたかどうか判断するポート増加判定を行ない、増加したポートを介してプリンタ名を取得し、取得したプリンタ名がインストーラの対応するプリンタか判断し、対応機種と判断された場合インストール処理を行なうようにしているので、USBインタフェースのような、ポート名があらかじめ決まっていないインタフェースでもプリンタのような機器を接続した際に最適なドライバをインストールすることができる。

【0076】〔第3の実施形態〕本実施形態は先の実施形態2より効率的にポートの追加を判断する例である。ハードウェア構成は上記実施形態と同様であるものとし、以下、図16と図17のフローチャートを用いて本実施形態におけるインストーラの動作を説明する。

【0077】プリンタインストーラの実行ファイルの名称は前述の第2実施形態同様にSETUP.EXEであり、ハードディスクに格納された状態で、あるいは前述の実施形態同様にCD-ROMやフロッピーディスクなどのメディアにより供給される。

【0078】これらのメディア上のインストーラのバイナリ（SETUP.EXE）がユーザにより明示的に起動されるか、あるいはウィザード経由で起動されると、図16のステップS201においてUSBドライバに関係するファイルをシステムにコピーする。USBドライバのファイルをあらかじめコピーすることで、ユーザはファイルを指定する必要がなくなる。

【0079】ステップS202では図7のインストーラ画面を表示する。この画面でユーザにUSBプリンタを繋ぐよう促す。ユーザがUSBプリンタを繋ぐと、ステップS203において、システムから送られるメッセージを待つ処理を行なう。ここで、メッセージとはシステムからアプリケーションに対して送られる命令である。例えば画面を書きなおすメッセージやキーが押された時にメッセージが送られる。

【0080】ステップS204では、システムから送られたメッセージがWM_DEVICECHANGEメッセージかどうか判断する。このWM_DEVICECHANGEメッセージは、パソコンと繋がっているデバイスの状態が変更されたときに送られる。ここでメッセージがWM_DEVICECHANGEメッセージならステップS205へ進み、そうでなければステップS203へ戻る。

【0081】ステップS205では、WM_DEVICECHANGEメッセージの要因を調べ、ポートが追加されたか否かを判断する。ポートが追加されたらステップS206へ進む。そうでなければステップS203へ戻る。

【0082】ステップS206では、追加されたポート

の名称の先頭が“USBPRN”かどうか判断する。ステップS206が肯定された場合にはステップS207へ進み、否定された場合にはステップS203へ戻る。ステップS207はステップS206でUSBと判断したポートをオープンするステップである。ステップS208はオープンしたポートからUSBプロトコルのGET_DEVICE_IDリクエストを要求し、デバイスIDを取得する。このデバイスIDは先の実施形態2と同じものである。ステップS209はステップS207でオープンしたUSBインタフェースのポートをクローズする。

【0083】ステップS210では、デバイスIDからプラグアンドプレイID(略してPnPID)を作る。前述の通り、PnPIDはデバイスIDのMFGタグの名称とMDLタグの名称を足したものである。

【0084】ステップS211では、インストーラが持つ対応プリンタリストの中にステップS210で作成したPnPIDが記載されているか否かを判断する。この対応プリンタリストは先の実施形態2と同じものである。ステップS210で作成したPnPIDが対応プリンタリストに記載されていれば対応プリンタと判断しステップS212へ進み、記載されていなければステップS203へ戻る。

【0085】ステップS212では、ステップS202で表示したインストール画面1を消去し、ステップS213では図8のインストール画面2を表示する。そして、ステップS214では対応プリンタリストに記載されているファイル名を取得する。さらに、ステップS215ではステップS214で取得したファイルをコピーするステップである。

【0086】ステップS216は対応プリンタリストのファイルを全てコピーしたか判断し、全てコピーしたらステップS217へ進む。ステップS217はステップS215でコピーしたファイルをプリンタドライバとしてシステムに登録する。さらに、ステップS218では、ステップS213で表示したインストール画面2を消去する。

【0087】以上のようにして、デバイス接続時のシステムから送られるメッセージが、WM_DEVICECHANGEメッセージかどうか判断し、さらにその内容に応じて入出力ポートが増えたかどうか判断するポート増加判定を行ない、増加したポートを介してプリンタ名を取得し、取得したプリンタ名がインストーラの対応するプリンタか判断し、対応機種と判断された場合インストール処理を行なうようにしているので、USBインタフェースのような、ポート名があらかじめ決まっていな

率的にポートの追加を判断することができる。

【0088】[第4の実施形態]以下に、さらに異なる情報を用いて最適なプリンタを判断する例を示す。本実施形態で用いる情報はUSBデバイスが必ず持っている情報なので、プリンタ以外のデバイスのインストーラに応用することも可能である。ハードウェア構成は上記実施形態2および3と同様であるものとし、図18と図19のフローチャートを用いて本実施形態におけるインストーラの動作を説明する。

10 【0089】プリンタインストーラの実行ファイルの名称は前述の第3実施形態同様にSETUP.EXEであり、ハードディスクに格納された状態で、あるいは前述の実施形態同様にCD-ROMやフロッピーディスクなどのメディアにより供給される。

【0090】これらのメディア上のインストーラのバイナリ(SETUP.EXE)がユーザにより明示的に起動されるか、あるいはウィザード経由で起動されると、図18のステップS301に進む。

20 【0091】ステップS301ではUSBドライバに係するファイルをシステムにコピーする。USBドライバのファイルをあらかじめコピーすることで、ユーザはファイルを指定する必要がなくなる。

【0092】ステップS302ではシステムが持つ全てのポート名を列挙し、ポートリストAとしてメモリに記憶する。ポートリストの構造は図9に示した通りで、ここでは図9はLPT1とCOM1とFILEとUSBPRN01というポートが記憶されている。

30 【0093】ステップS303は図7のインストーラ画面を表示するステップである。この画面でユーザにUSBプリンタを繋ぐよう促す。そして、ステップS304でポートチェックのインターバル時間になるまで待つ処理である。本実施形態では第2実施形態同様にインターバル時間を2秒とし、2秒になったらステップS305へ進む。

40 【0094】ステップS305ではシステムが持つ全てのポート名を列挙し、ポートリストBとしてメモリに記憶する。ステップS306では、ポートリストBからポートリストAに存在するポートを削除するステップである。削除した結果、ポートリストBには新たに追加されたポートが残る。もし追加されていなければ何も残らない。

【0095】ステップS307はポートリストBにポートが残っているか判断するステップである。ポートが残っていなければステップS304へ戻る。また、ポートが残っていればステップS308へ進む。

50 【0096】ステップS308はポートリストBに残っているポート名がUSBインタフェースのポートか判断するステップである。本実施形態ではUSBインタフェースのポート名は“USBPRN数字”となるので、ポートリストBに残っているポート名の先頭が“USBP

RN”かどうか判断し、先頭が“USBPRN”ならUSBインタフェースのポートであると判断する。USBインタフェースのポートの場合はステップS309へ進み、そうでなければステップS304へ戻る。

【0097】ステップS309はUSBインタフェースのポートをオープンする。ステップS310では、オープンしたポートからUSBプロトコルを用いてGET_DESCRIPTORリクエストを要求し、デバイスデスクリプタを取得する。このデバイスデスクリプタはデバイスの基本的な情報を格納した構造体であり、USBデバイスに限らず全てのWindowsデバイスファイルに共通するデータである。デバイスデスクリプタの構造とその内容の例を図12に示す。図12において、デバイスデスクリプタのベンダーIDは製造者を意味するIDで、プロダクトIDは製品を区別するIDである。図12の場合ベンダーIDは04A9でプロダクトIDは0000となっている。

【0098】ステップS311では、ステップS309でオープンしたUSBインタフェースのポートをクローズする。そして、ステップS312では、ステップS310で取得したデバイスデスクリプタからベンダーIDとプロダクトIDを取り出す。

【0099】ステップS313では、インストーラが持つ対応プリンタリストの中にステップS312で取り出したベンダーIDとプロダクトIDが記載されているか判断する。本実施形態のインストーラが有する対応プリンタリストの構造を図13に示す。対応プリンタリストはベンダーIDとプロダクトIDと対応するファイル名から構成される。もしステップS312で作成したベンダーIDとプロダクトIDが対応プリンタリストに記載されていれば対応プリンタと判断しステップS314へ進み、記載がなければステップS304へ戻る。

【0100】ステップS314はステップS303で表示したインストール画面1を消去する。そして、ステップS315において図8のインストール画面2を表示する。インストール画面2はファイルのコピー中表示される。さらに、ステップS316は対応プリンタリストに記載されているファイル名を取得し、ステップS317はステップS316で取得したファイルをコピーする。

【0101】ステップS318は対応プリンタリストのプリンタのデバイスドライバのファイルを全てコピーしたか判断する。全てのコピーが終了したらステップS319へ進み、そうでなければステップS317に戻る。そして、ステップS319ではステップS317でコピーしたファイルをプリンタドライバとしてシステムに登録し、ステップS320ではステップS315で表示したインストール画面2を消去する。

【0102】以上のようにして、GET_DESCRIPTORプロトコルを用いてデバイスデスクリプタを取得し、デバイス名（プリンタ名）を取得し、入出力ポ

ートが増えたかどうか判断するポート増加判定を行ない、増加したポートを介してプリンタ名を取得し、取得したプリンタ名がインストーラの対応するプリンタか判断し、対応機種と判断された場合インストール処理を行うようにしているので、USBインタフェースのような、ポート名があらかじめ決まっていないインタフェースでもプリンタのような機器を接続した際に最適なドライバをインストールすることができる。

【0103】本実施形態はUSBインタフェースに限定したものであるが、デバイスドライバがインストールされるとポートが追加されるインタフェースであれば、どのようなインタフェースを用いても実施できる利点がある。

【0104】以上では、OSとしてMicrosoft社のWindowsを考えたが、任意のOSにおいてプリンタなどのデバイスを特定するのに類似のメカニズムを用いている場合は、上記の技術を応用できるのはいうまでもない。

【0105】[第5の実施形態] 本実施形態では、USBポート、イーサネットなどのインタフェースを介して複数のプリンタなどの外部機器を接続する場合に、自分が使用したい外部機器を容易に確認できるようにするための構成を例示する。

【0106】本実施形態では、ホストで選んだプリンタに「確認コマンド」を送信し、それにプリンタを応答させることにより自分が使用したいプリンタを容易に確認できるようにする。より具体的には、「確認コマンド」を受けたプリンタは可聴音を発生し、これにより、ユーザがどのプリンタから音が鳴ったか判断する構成である。これにより、ホストで表示しているプリンタと実際のプリンタとの対応付けを確認することができる。

【0107】最初に本実施形態を実現するためのハードウェア構成を図20を用いて説明する。図20は、図1あるいは図6と同等の図面で、同一ないし相当する部材には同一符号を付してある。プリンタインタフェースH07はここでもUSBポートのような複数台のプリンタH81、H82…を同時接続できるようなポートであるものとする。それ以外の構成は図1あるいは図6と同じであるので、ここでは詳細な説明を省略する。

【0108】本実施形態においては、コンピュータ（ホスト）からプリンタH81、H82…に「確認コマンド」を送信するが、その際に図21のフローチャートのような制御手順と、図22のようなダイアログを用いる。

【0109】図21の制御手順は、CPU H01が実行可能なプログラムとしてハードディスクに格納された状態で、あるいは前述の実施形態同様にCD-ROMやフロッピーディスクなどのメディアにより供給される。より具体的には、このようなプログラムはプリンタドライバプログラムの一部として供給し、コントロールパネ

ルやプリンタ制御用のメニューやアイコンから、あるいはアプリケーションのプリントダイアログ経由で起動されるものとする。メニューやアイコンのタイトルは「プリンタ確認」「プリンタ選択」などのものが考えられる。

【0110】上記のようなメニューやアイコンにより起動されると、同プログラムは、図21のステップS01において、ホストと接続しているプリンタ名をリスト表示する。図22の真ん中左側がホストと接続しているプリンタ名をリストするための表示領域で、プリンタ数が多ければこの領域はスクロール可能に制御される。プリンタ名は、図12に示したようなGET_DESCRIPTORプロトコルを用いて取得したプロダクト名文字列を用いることができる他、同一のプリンタが存在する場合にはさらにこのプロダクト名文字列の後に自動的に番号を付与して生成した文字列を用いたり、あるいはユーザが任意に設定したニックネームを用いることも考えられる。

【0111】ステップS02では、ユーザからのマウスクリック待ちを行なう。ここでは、ユーザは所望のプリンタ名をマウスでクリックすることにより、選択し、「確認」ボタンを押下することにより確認コマンドの送信を行なう。また、「選択」ボタンを押下することにより印刷に用いるプリンタを選択することができる。

【0112】クリックイベントが発生した場合には、ステップS03において、図22のダイアログのプリンタ名の部分がクリックされたか否かを判断する。プリンタ名がクリックされると、ステップS04とステップS05が実行される。すなわち、ステップS04は今まで反転表示していたプリンタ名を元に戻し、ステップS05では新たにクリックされたプリンタ名を反転表示する。

【0113】ステップS06、S08、S10はそれぞれ図22のダイアログの「確認」、「キャンセル」、「選択」の各ボタンのクリックを検出するものである。

【0114】まず、ステップS06では、図22の「確認」ボタンが押されたか否かを判断し、「確認」ボタンが押されると、ステップS07において、現在反転表示中のプリンタに対して確認コマンドを送る。この確認コマンドのフォーマットは任意であり、本実施形態ではプリンタのスピーカなどを用いて可聴音を発生させることができればどのようなものであってもよい。たとえばそのプリンタのPDL（ページ記述言語）などに対応するコマンド（警告音発生のためのコマンドなど）があればそれを用いることができる。プリンタはこのコマンドを受信するとピープ音を鳴らす。

【0115】ステップS08で図22の「キャンセル」ボタンが押された場合は、ステップS09において全てのプリンタ選択を無効としダイアログの処理を終了する。ステップS10で「選択」ボタンが押された場合は、ステップS11で現在反転表示しているプリンタを

新たに選択したプリンタとしてダイアログの処理を終了する。ステップS10で「選択」ボタンが押されていない場合はステップS02へ戻り新たなマウスクリックを待つ。

【0116】一方、プリンタ側では、図23のフローチャートのような処理を行なう。すなわち、ステップS01ではコマンドの受信待ちを行ない、コマンドを受信した場合はステップS02で受信したコマンドが図21のステップS07で送信された確認コマンドかどうかを判断し、確認コマンドだった場合、ステップS03でピープ音を鳴らす。

【0117】ステップS04以後は本来のプリンタの処理であり、受信コマンドが印刷コマンドであれば対応する印刷処理を行なう（ステップS05）。

【0118】このようにして、ユーザはダイアログ（図22）を用いてプリンタのリストを調べ、目的のプリンタに確認コマンドを送信してピープ音を発生させることができるので、自分が使用したいプリンタを容易に確認することができる。

【0119】上記の技術は、プリンタに限らず、USBポート、イーサネットなどのインタフェースを介して複数の外部機器（たとえばスキャナなどの入力機器であってもよい）を接続する場合に、自分が使用したい外部機器を容易に確認できるようにするために用いることができるものである。

【0120】なお、本実施形態では確認コマンドとして音を鳴らすコマンドを用いたが、本発明の本質は印刷と異なる動作をするコマンドをプリンタに実行させることで、ユーザが印刷先のプリンタを確認できるように報知することであるから、確認コマンドに応じてプリンタで実行させる動作は可聴音の発生だけではなく、例えばLEDを点滅させるような動作であってもよい。

【0121】〔第6の実施形態〕上記の第5実施形態とは逆に、外部機器（プリンタ）側で操作を行なうことにより、その外部機器を出力（あるいは入力）に用いるべく「選択」状態とすることも考えられる。

【0122】たとえば、図26に示すように、外部機器としてのプリンタ（図20のH81あるいはH82に相当する）に選択ボタンB02を設けておき、ユーザがそのプリンタを出力に用いたい場合、この選択ボタンB02を押下する。この選択ボタンB02の押下は所定のプロトコルにより、ホスト側に伝達され、これを受信したホスト側ではそのプリンタを出力に用いるべく「選択」状態に制御する。なお、図26のボタンB01は電源ボタンである。

【0123】本実施形態のホスト（コンピュータ）側のハードウェア構成は第5実施形態と同一でよい。本実施形態においては、図24のフローチャートのような制御手順と図25のようなダイアログを用いる。図24のプログラムは、第5実施形態と同様の契機で起動すればよ

い。図24の処理の全体構成は図21とほぼ同様であるので、以下では図21の処理と同様の細部については説明を省略する。

【0124】図24のステップS01では、現在接続しているプリンタを図25のダイアログに表示する。

【0125】ステップS02ではマウス入力待ち、マウス入力があった場合はまずステップS03でステップS01で表示した全プリンタのステータスを得る。このステータスには図26の選択ボタンB02が押されているか否かを示す「選択ステータス」が含まれているものとする。前述のように、この選択ボタンB02の押下は所定のプロトコルによりホスト側に伝達される。この選択ステータスが1なら当該のプリンタで選択ボタンB02が押されたことを意味している。

【0126】ステップS04では「選択ステータス=1」のステータスを返したプリンタがあるかを判断し、選択ステータス=1のプリンタがあればステップS05へ進み、今まで反転表示していたプリンタ名の表示を元に戻し、さらにステップS06で選択ステータス=1となっていたプリンタ名の表示を反転する。

【0127】ステップS07はステップS02で得たマウス入力情報からマウスクリックが発生したか判断し、マウスクリックが発生したらステップS08へ進み、マウスクリックが発生していなければステップS02へ戻る。

【0128】ステップS08は図25のダイアログのプリンタ名をクリックされたか判断し、クリックされていればステップS09へ進み、クリックされていなければステップS11へ進む。

【0129】ステップS09では、今まで反転表示していたプリンタ名の表示を元に戻し、ステップS10ではマウスクリックされたプリンタ名の表示を反転するステップである。すなわち、ここでは図21同様に手動操作によりプリンタの選択を行なえるようにしている。

【0130】ステップS11では、図25のダイアログの「キャンセル」ボタンが押されたかを判断する。同ボタンが押されていればステップS12へ進み、プリンタの選択を全て無効とし、ダイアログの処理を終了する。

【0131】ステップS13では、図25の「選択」ボタンが押されたか否かを判断し、押されていなければステップS02へ戻り新たなマウス入力を受け付ける。一方、確認ボタンが押された場合はステップS14において現在反転表示しているプリンタを新たに選択したプリンタとしてダイアログの処理を終了する。

【0132】プリンタ側では、図27のフローチャートのような処理を行なうことにより選択ステータスを決定することができる。図27は選択ステータスに関する処理を行うタスクを説明するものであり、プリンタ本来の処理は別のタスクとして動作させればよいので、ここではプリンタ本来の処理については説明を省略する。

【0133】図27のステップS01はプリンタの操作ボタンの押下を受け付け、ステップS02では選択ボタン(図26のB02)が押されたか判断する。選択ボタンが押された場合ステップS03へ進み、選択ステータスを1に設定し、選択ボタンが押されていない場合にはステップS04において選択ステータスを0に設定する。プリンタはこのステータスを保持し、ステータス要求があったときに所定のプロトコルを用いてホストへ送信すればよい。

【0134】このようにして、ユーザはダイアログ(図25)を用いてプリンタのリストを調べた時、プリンタ側で選択ボタンを押下していれば既にそのプリンタが反転表示状態となっているので、単にその選択状態を「選択」ボタンにより追認するだけで自分が使用したいプリンタを容易に確認し、選択することができる。

【0135】本実施形態の技術も、プリンタに限らず、USBポート、イーサネットなどのインタフェースを介して複数の外部機器(たとえばスキャナなどの入力機器であってもよい)を接続する場合に、自分が使用したい外部機器を容易に確認できるようにするために用いることができるものである。

【0136】本実施形態では、専用の「選択」ボタンをプリンタに設ける構成を例示したが、例えばオンラインボタンや他の既存の操作ボタン、あるいは複数の既存の操作ボタンの押下の組合せを「選択」操作に用いるようにしてもかまわない。

【0137】[第7の実施形態]あるいは、本実施形態はプリンタの電源を入れるとホストのプリンタドライバで自動的にそのプリンタを印刷先として「選択」させることも考えられる。本実施形態のハードウェア構成は実施形態1と同じでよい。

【0138】本実施形態においては、図28のフローチャートのような制御手順と図25のダイアログ(第6実施形態と同じ)を用いる。図28のプログラムは、第5、6実施形態と同様の契機で起動すればよい。図28の処理の全体構成は図21、24とほぼ同様であるので、以下では図21、24の処理と同様の細部については説明を省略する。

【0139】図28のステップS01では現在接続しているプリンタを図25のダイアログで表示し、ステップS02でマウス入力待ち。

【0140】ステップS03では新たに接続されたプリンタがあるかどうか判断する。ここで、「接続」とはUSBバスや、イーサネットなどのネットワーク上でプリンタが見えるようになればよい。そのためUSBバスや、イーサネットなどを介して接続されたプリンタが電源オフの状態から電源オンの状態に変わったときもこのステップで検出できる。具体的には、バスあるいはネットワーク上のプリンタのアドレスを監視したり、図26の電源ボタンB01などにより電源が投入された時にプ

リント側から明示的な信号を送信するようにしても、新たに「接続」されたプリンタがあるかどうかを判断することができる。

【0141】新たに接続されたプリンタがあった場合はステップS04において、今までダイアログ上で反転表示していたプリンタ名の表示を元に戻し、ステップS05で新たに「接続」されたプリンタのプリンタ名の表示を反転する。

【0142】ステップS06では、ステップS02で得たマウス入力情報からマウスクリックが発生したか判断し、マウスクリックが発生したらステップS07へ進み、発生していなければステップS02へ戻る。

【0143】ステップS07では図25のダイアログのプリンタ名がクリックされたかを判断し、クリックされていればステップS08へ進み、クリックされていなければステップS10へ進む。

【0144】ステップS08では今まで反転表示していたプリンタ名の表示を元に戻し、ステップS09ではマウスクリックされたプリンタ名の表示を反転する。

【0145】ステップS10は図25のダイアログの「キャンセル」ボタンが押されたかを判断し、押されていればステップS11へ進む。ステップS11ではプリンタの選択を全て無効とし、ダイアログの処理を終了する。

【0146】ステップS12では図25のダイアログの「選択」ボタンが押されたかどうかを判断し、押されていなければステップS02へ戻り新たなマウス入力待つ。確認ボタンが押された場合にはステップS13において現在反転表示しているプリンタを新たに選択したプリンタとしてダイアログの処理を終了する。

【0147】このようにして、ユーザはダイアログ（図25）を用いてプリンタのリストを調べた時、電源投入したプリンタが反転表示状態となっているので、単にその選択状態を「選択」ボタンにより追認するだけで自分が使用したいプリンタを容易に確認し、選択することができる。

【0148】本実施形態の技術も、プリンタに限らず、USBポート、イーサネットなどのインタフェースを介して複数の外部機器（たとえばスキャナなどの入力機器であってもよい）を接続する場合に、自分が使用したい外部機器を容易に確認できるようにするために用いることができるものである。

【0149】

【発明の効果】以上の説明から明らかなように、本発明によれば、周辺機器デバイスを当該機器のデバイスドライバ経由で制御することにより当該機器に対する入出力を行なうとともに、デバイスドライバの登録に関するレジストリ情報がレジストリ情報記憶手段に登録される電子機器、その制御方法、およびその制御プログラムを格納したコンピュータ読取可能な記憶媒体において、所定

の周辺機器のデバイスドライバをシステムにインストールする際、登録すべきデバイスドライバに対応したデバイスID情報を前記レジストリ情報記憶手段で検索し、検索されたレジストリ情報の一部が不明となっていた場合、当該レジストリ情報を削除する構成を採用しているため、明示的ドライバを削除するような操作を必要とせず、また、高額なユーザサポートを必要とせずに容易に自分自身でUSBデバイスのインストールを行なうことができる、という優れた効果がある。

【0150】あるいは、複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続し、周辺機器デバイスを当該機器のデバイスドライバ経由で制御することにより当該機器に対する入出力を行なう電子機器、その制御方法、およびその制御プログラムを格納したコンピュータ読取可能な記憶媒体において、前記入出力ポートが増加したかどうか判断し、前記入出力ポートが増加した場合に増加したポートを介してそのポートに対応する周辺機器のデバイス名を取得し、取得したデバイス名が特定の対応機種を示している場合に当該周辺機器のデバイスドライバのインストール処理を行なう構成を採用することにより、USBインタフェースのように、ポート名があらかじめ決まっておらず、複数の周辺機器を同時接続可能な入出力ポートでも、プリンタのような機器を接続した際に最適なドライバをインストールすることができる、という優れた効果がある。

【0151】あるいは、複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続して当該機器に対する入出力を行なう電子機器、その制御方法、およびその制御プログラムを格納したコンピュータ読取可能な記憶媒体において、ユーザが入出力先として選択すべき周辺機器を確認するための確認コマンドを前記入出力ポート経由で送信し、該確認コマンドを受信した周辺機器側において該確認コマンドの受信をユーザに報知する構成を採用することにより、USBインタフェースのように、ポート名があらかじめ決まっておらず、複数の周辺機器を同時接続可能な入出力ポートでも、ユーザは自分の使用すべきプリンタなどの周辺機器を容易に特定することができる、という優れた効果がある。

【0152】あるいは、複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続して当該機器に対する入出力を行なう電子機器、その制御方法、およびその制御プログラムを格納したコンピュータ読取可能な記憶媒体において、前記周辺機器の操作または動作状態に応じて定まるステータス情報を前記入出力ポートを介して前記周辺機器デバイスから受信し、受信したステータス情報に応じて入出力先として周辺機器を選択する構成を採用することにより、周辺機器の操作または動作状態、たとえば、周辺機器の操作手段の操作状態や主電源の投入状態などに応じて定まるステータスを用いて自分が使用したいプリンタなどの周辺機器を容易

に確認し、選択することができる、という優れた効果が得られる。

【図面の簡単な説明】

【図 1】本発明を実施可能なパーソナルコンピュータのハードウェア構成を示したブロック図である。

【図 2】本発明によるデバイスドライバのインストール処理に用いられるファイルの構成を示した説明図である。

【図 3】図 2 の USBDEV. INF の内容を示した説明図である。

【図 4】レジストリエディタによりシステムのレジストリをオープンした状態を示した説明図である。

【図 5】本発明の第 1 実施形態における制御手順を示したフローチャート図である。

【図 6】本発明を実施可能なパーソナルコンピュータのハードウェア構成を示したブロック図である。

【図 7】本発明において表示されるインストーラ画面 1 を示した説明図である。

【図 8】本発明において表示されるインストーラ画面 2 を示した説明図である。

【図 9】本発明において用いられるポートリストを示した説明図である。

【図 10】本発明において用いられるデバイス ID を示した説明図である。

【図 11】本発明のインストーラプログラムが有する対応プリンタリストを示した説明図である。

【図 12】本発明において用いられるデバイスディスクリプタを示した説明図である。

【図 13】本発明のインストーラプログラムの異なる対応プリンタリストを示した説明図である。

【図 14】本発明の第 2 実施形態における制御手順を示したフローチャート図である。

【図 15】本発明の第 2 実施形態における制御手順を示したフローチャート図である。

【図 16】本発明の第 3 実施形態における制御手順を示

したフローチャート図である。

【図 17】本発明の第 3 実施形態における制御手順を示したフローチャート図である。

【図 18】本発明の第 4 実施形態における制御手順を示したフローチャート図である。

【図 19】本発明の第 4 実施形態における制御手順を示したフローチャート図である。

【図 20】本発明を実施可能なパーソナルコンピュータのハードウェア構成を示したブロック図である。

10 【図 21】本発明の第 5 実施形態における制御手順を示したフローチャート図である。

【図 22】本発明において用いられるプリンタ選択ダイアログを示した説明図である。

【図 23】プリンタ側の制御を示したフローチャート図である。

【図 24】本発明の第 6 実施形態における制御手順を示したフローチャート図である。

【図 25】本発明において用いられるプリンタ選択ダイアログを示した説明図である。

20 【図 26】選択ボタンを設けたプリンタの構成を示した説明図である。

【図 27】プリンタ選択ステータスを示したフローチャート図である。

【図 28】本発明の第 7 実施形態における制御手順を示したフローチャート図である。

【符号の説明】

H01 CPU

H02 キーボード

H03 表示器

30 H04 ROM

H05 RAM

H06 ハードディスク

H07 USB インタフェース

H08 USB デバイス

【図 2】

(図 2)

ファイル名	説明
USBDEV. SYS	デバイスドライバ本体
USBDEV. INF	設定ファイル
SETUP. EXE	インストーラプログラム

ファイル構成図

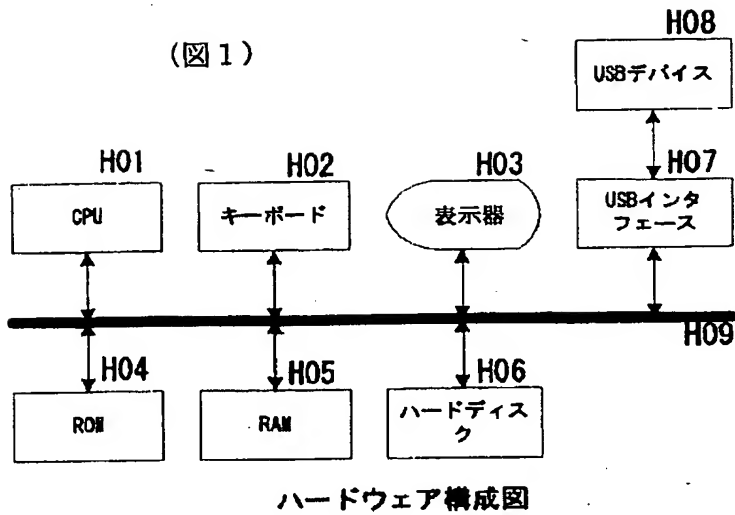
【図 9】

(図 9)

番号	ポート名
1	LPT1
2	COM1
3	FILE
4	USBPRN01

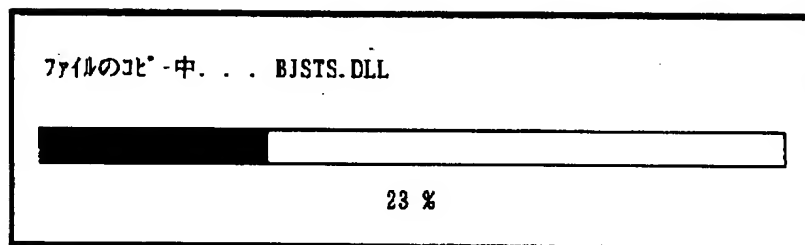
ポートリスト

【図1】



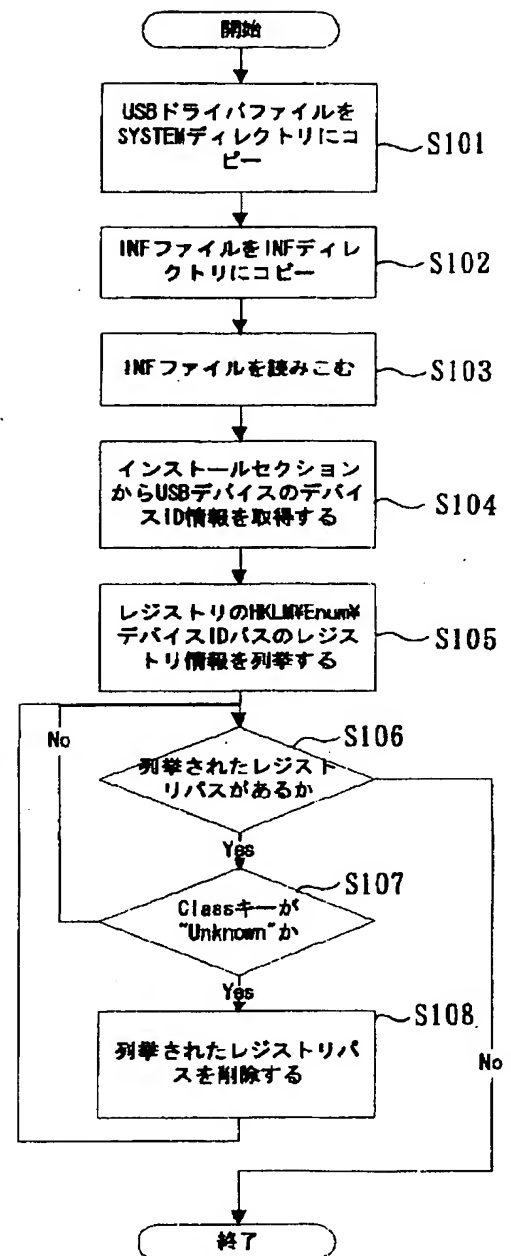
【図8】

(図8)



インストーラ画面2

【図5】



(図5)

【図10】

(図10)

MFG:Qanon;CMD:TEXT;MDL:ABC-3000;CLS:PRINTER;

デバイスID

【図3】

(図3)

```
1.  [Version]
2.  Signature="$Windows 95$"
3.  Class=USB
4.  Provider=Qanon

5.  [DestinationDirs]
6.  DefaultDestDir=11

7.  [Manufacturer]
8.  Qanon=Qanon.Section

9.  [Qanon.Section]
10. "Qanon USB Device"=Qanon.USB,USB\VID_04A9&PID_1052

11. [Qanon.USB]
12. CopyFiles=Qanon.CopyList
13. AddReg=Qanon.AddReg

14. [Qanon.CopyList]
15. usbdev.sys

16. [Qanon.AddReg]
17. HKR,,DevLoader,,*ntkern
18. HKR,,NTMPDriver,,usbdev.sys
19. HKR,,QanonSoft,,Printer
20. HKR,,Device,,USBPRN
21. HKR,,ID,,VID_04A9&PID_1052
22. HKR,,Instance,,Multiple
23. HKR,,DeviceDesc,, "Qanon USB Device"

24. [ControlFlags]
25. ExcludeFromSelect=USB\VID_04A9&PID_1052

26. [PreCopySection]
27. HKR,,NoSetupUI,,1

28. [SourceDisksNames]
29. 1 = "Qanon USB Device Disk",,0001

30. [SourceDisksFiles]
31. usbdev.sys = 1
```

USBDEV. INFの内容

【図4】

レジストリ エディタ	
名前	データ
<div> <input type="checkbox"/> Enum </div> <div> <input type="checkbox"/> BIOS <input type="checkbox"/> ESDI <input type="checkbox"/> FLOP <input type="checkbox"/> HTREE <input type="checkbox"/> INFRARED <input type="checkbox"/> ISAPNP <input type="checkbox"/> LPTENUM <input type="checkbox"/> MF <input type="checkbox"/> MONITOR <input type="checkbox"/> Network <input type="checkbox"/> PCI <input type="checkbox"/> Root <input type="checkbox"/> SCSI <input type="checkbox"/> USB <div> <input type="checkbox"/> ROOT_HUB <input type="checkbox"/> VID_047E&PID_1001 <input type="checkbox"/> VID_04A9&PID_1052 </div> </div>	<div> <input type="checkbox"/> (標準) <input type="checkbox"/> Capabilities <input type="checkbox"/> Class <input type="checkbox"/> CompatibleIDs <input type="checkbox"/> ConfigFlags <input type="checkbox"/> DeviceDesc <input type="checkbox"/> HardwareID </div> <div> (値の設定なし) 14 00 00 00 "Unknown" "USB&CLASS_07&SUBCLASS_01&PROT_02.USB&CLA 40 00 00 00 "Qanon USB Device" "USB&VID_04A9&PID_1052&REV_0103.USB&VID_0 </div>

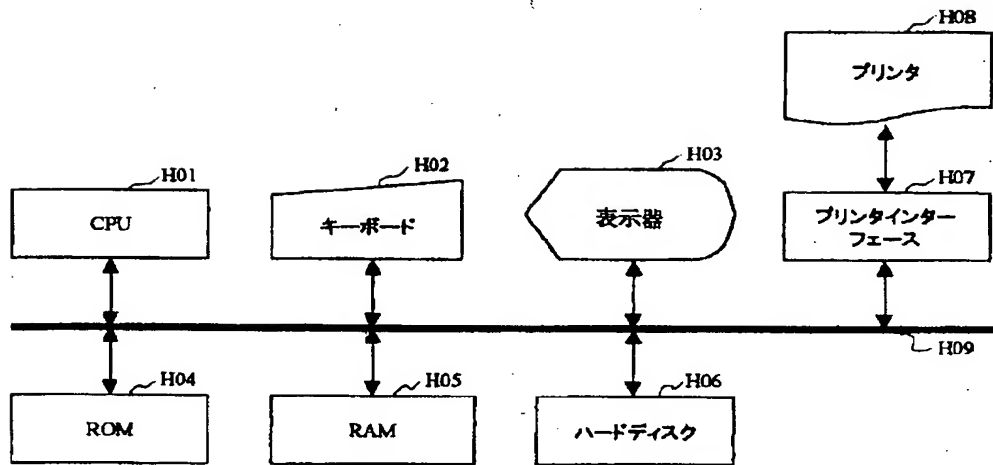
マイコンビュータ\\FREY\\LOCAL_MACHINE\\Enum\\USB\\VID_04A9&PID_1052\\0000000

レジストリ

(図4)

【図6】

(図6)



ハードウェアブロック図

【図11】

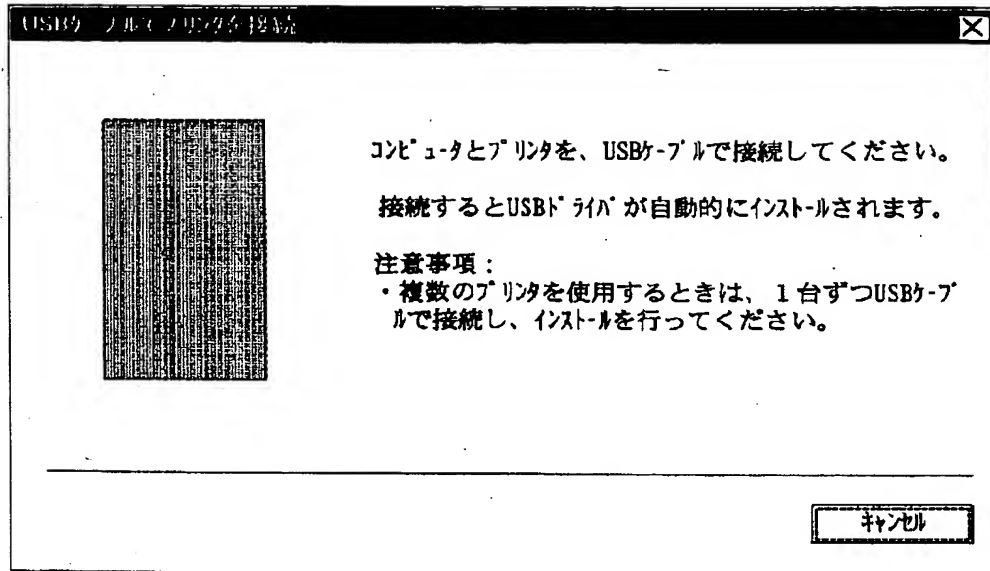
(図11)

番号	PnP ID	必要ファイル
1	QanonABC-3000	ABC3000.DRV ABC3000.DLL QANON_UI.DLL QANON_UN.EXE
2	QanonABC-80v	ABC80V.DRV ABC80V.DLL QANON_UI.DLL QANON_UN.EXE
3	QanonABC-8000	ABC8000.DRV ABC8000.DLL QANON_UI.DLL QANON_UN.EXE

対応プリンタリスト

【図7】

(図7)



インストーラ画面1

【図12】

(図12)

フィールド名	値(16進数)	意味
bLength	18	デバイスデスク립タのバイトサイズ
bDescriptorType	01	デバイスデスク립タのタイプ番号
bcdUSB	0100	USBバージョン番号
bDeviceClass	00	デバイスクラスコード
bDeviceSubClass	00	デバイスサブクラスコード
bDeviceProtocol	00	デバイスプロトコルコード
bMaxPacketSize	08	最大パケットサイズ
idVendor	04A9	ベンダーID
idProduct	0000	プロダクトID
bcdDevice	0100	デバイスバージョン番号
iManufacturer	01	ベンダー名文字列番号
iProduct	02	プロダクト名文字列番号
iSerialNumber	03	シリアル番号文字列番号
bNumConfigurations	01	コンフィグレーション数

デバイスデスク립タの構造

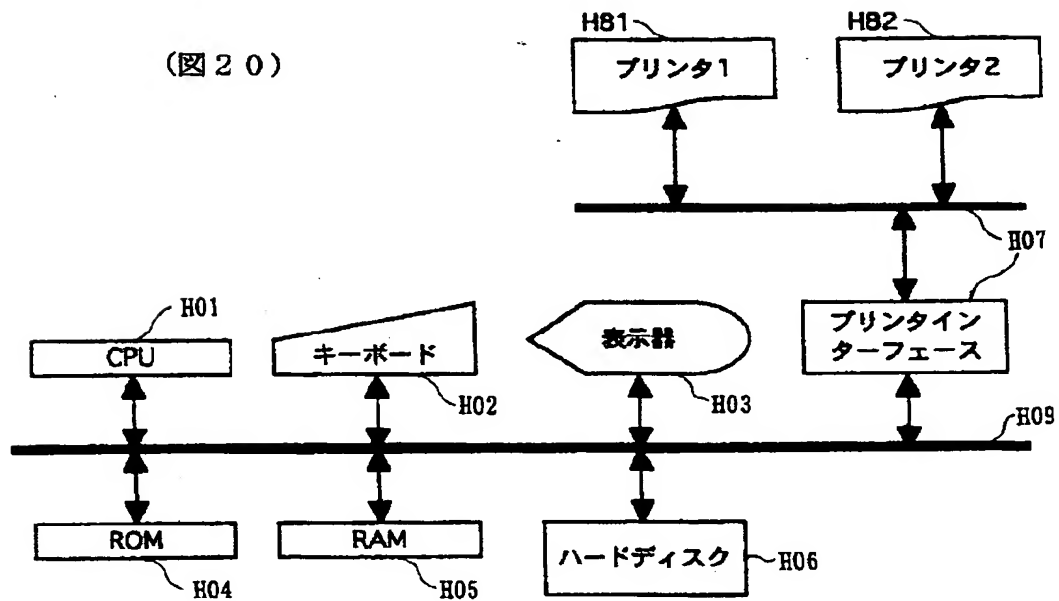
【図13】

(図13)

番号	ベンダーID	プロダクトID	必要ファイル
1	04A9	0001	ABC3000.DRV ABC3000.DLL QANON_UI.DLL QANON_UN.EXE
2	04A9	0002	ABC80V.DRV ABC80V.DLL QANON_UI.DLL QANON_UN.EXE
3	04A9	0003	ABC8000.DRV ABC8000.DLL QANON_UI.DLL QANON_UN.EXE

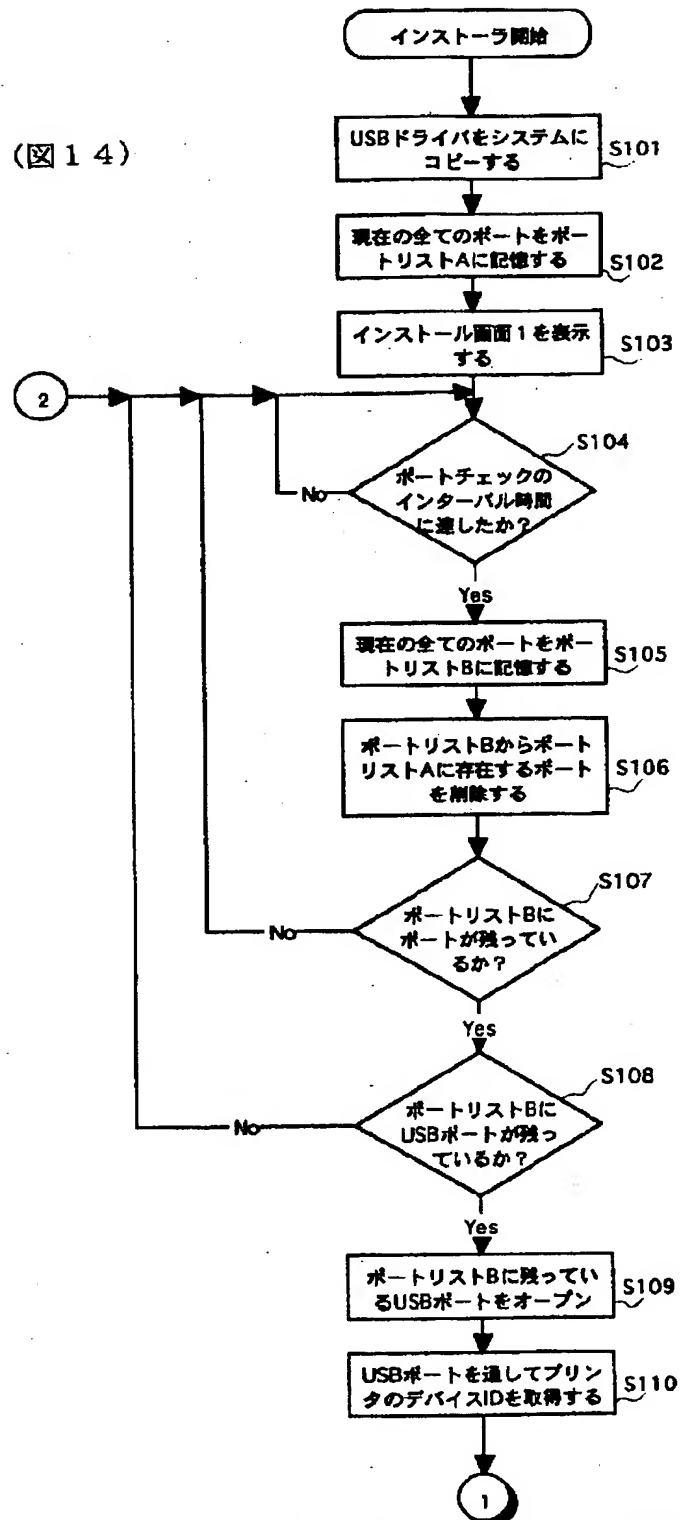
実施形態3の対応プリンタリスト

【図20】



ハードウェアブロック図

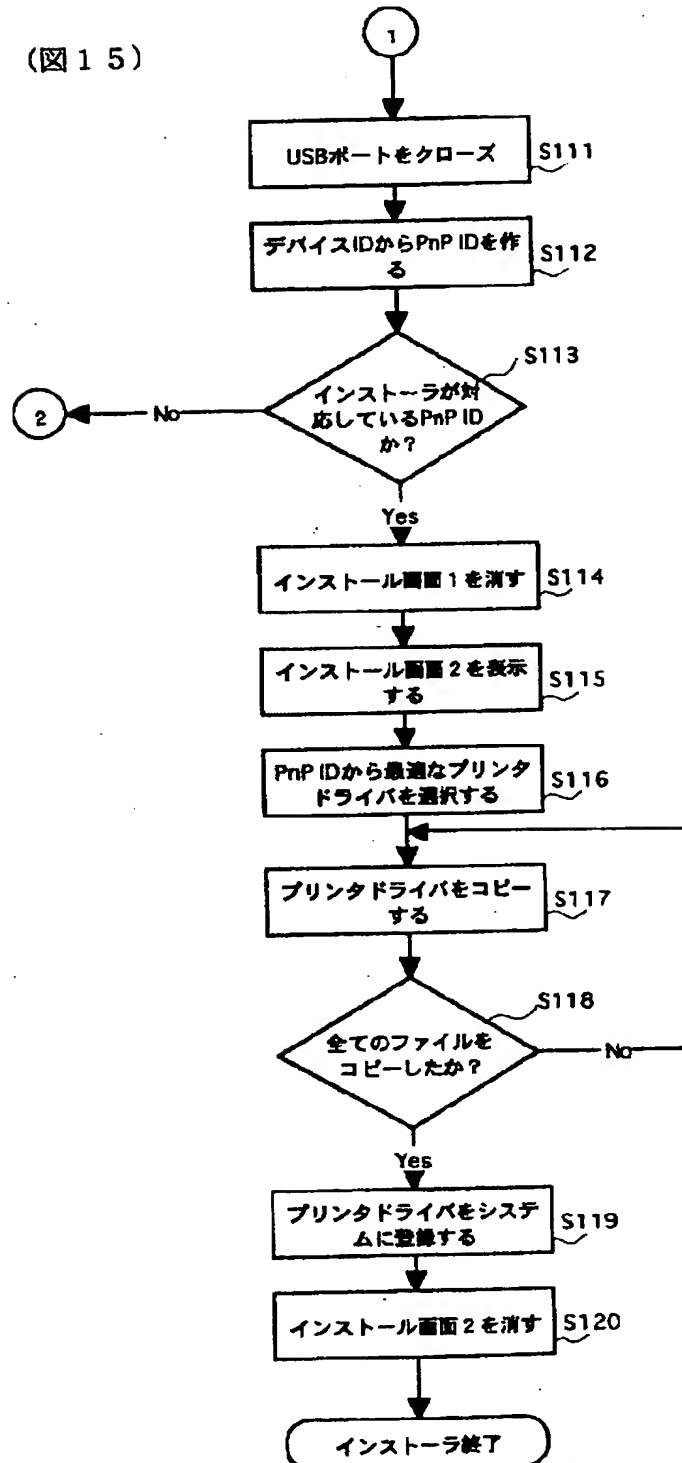
【図14】



実施形態2のフローチャート前半

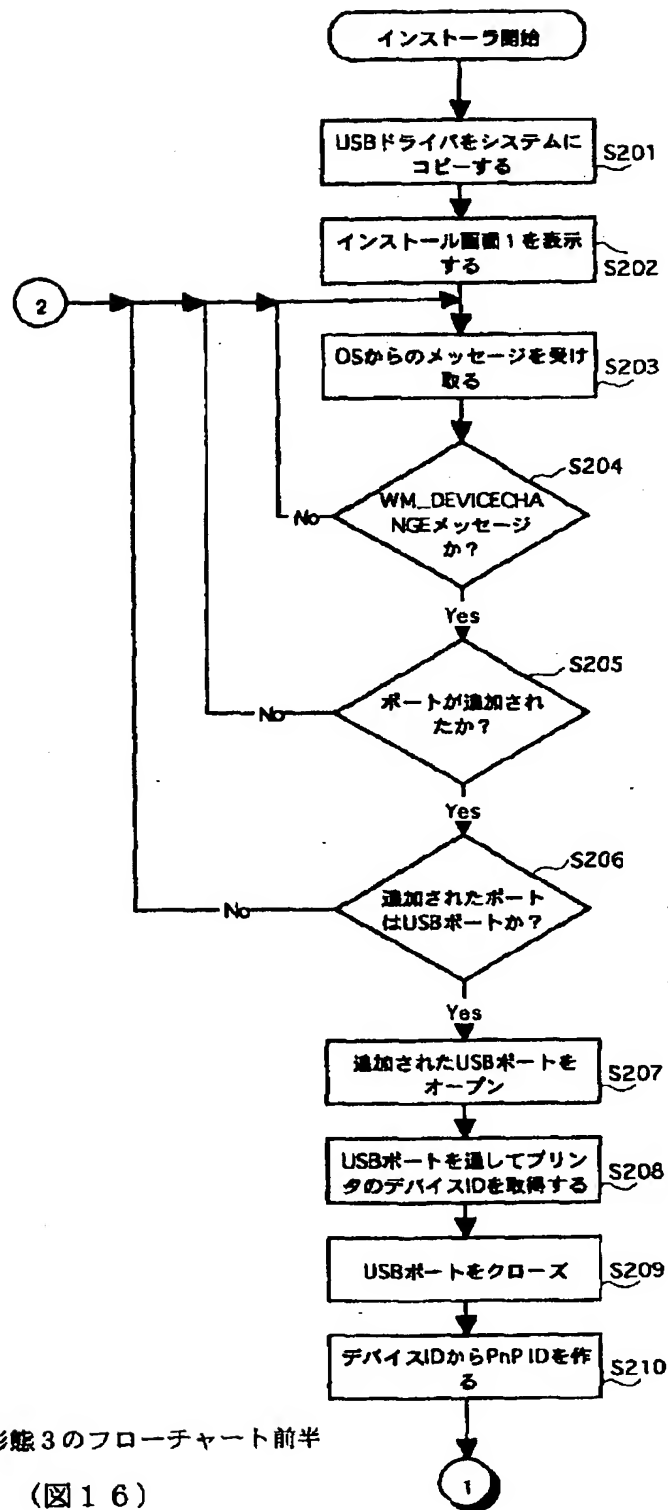
【図15】

(図15)



実施形態2のフローチャート後半

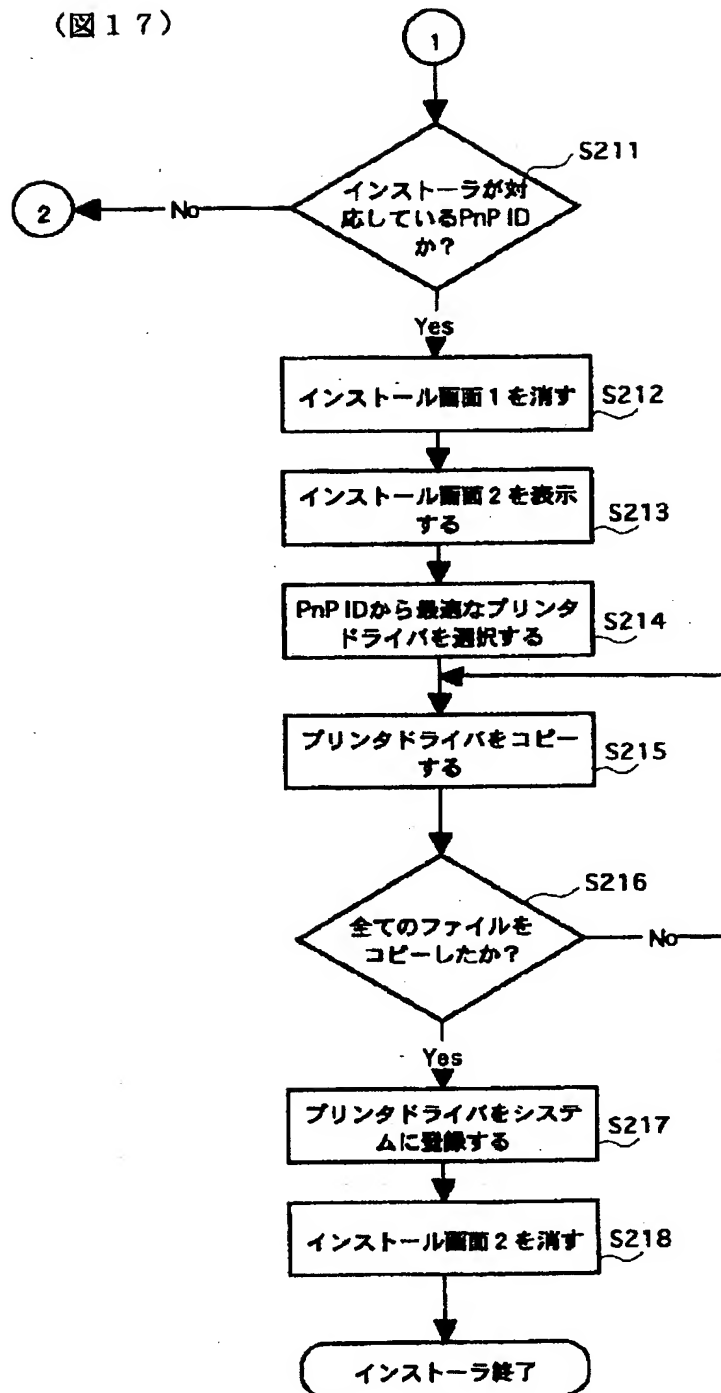
【図16】



実施形態3のフローチャート前半

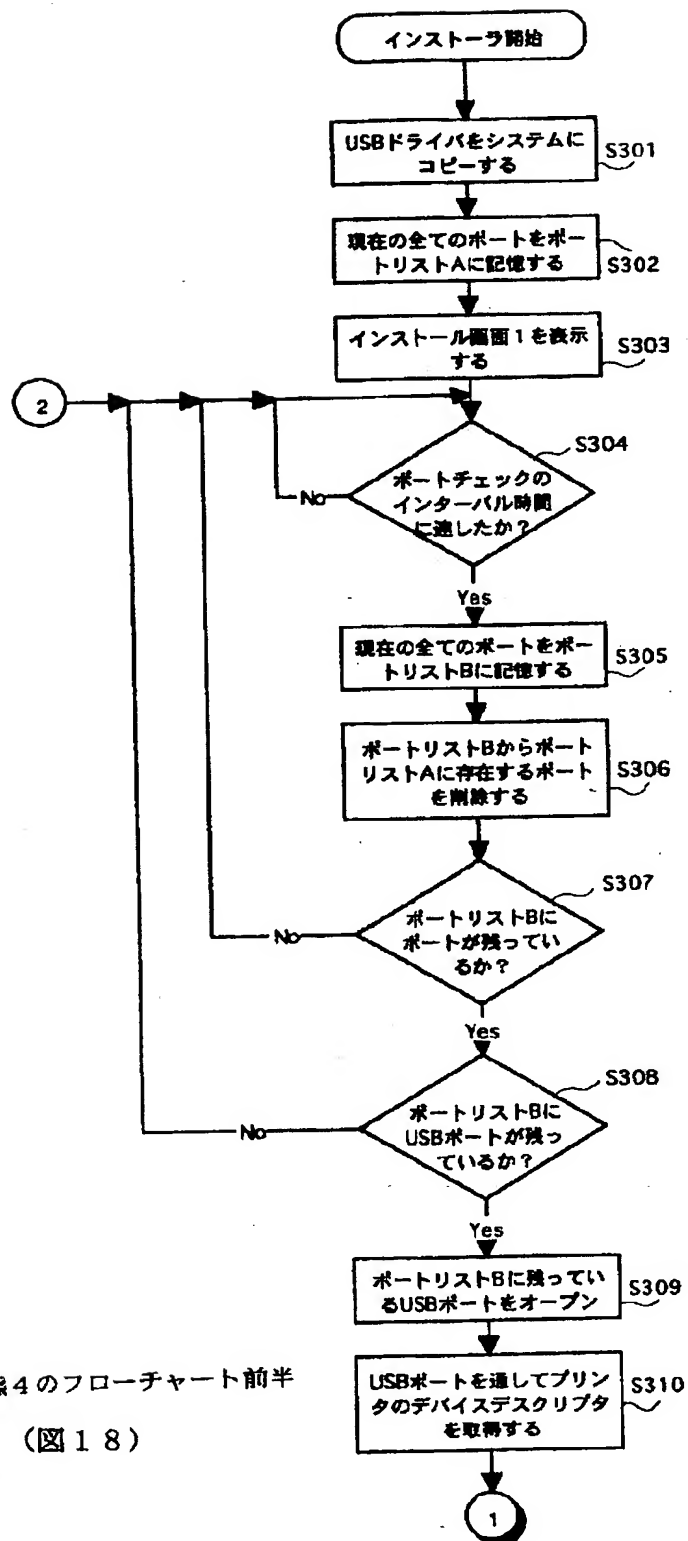
(図16)

【図17】



実施形態3のフローチャート後半

【図18】

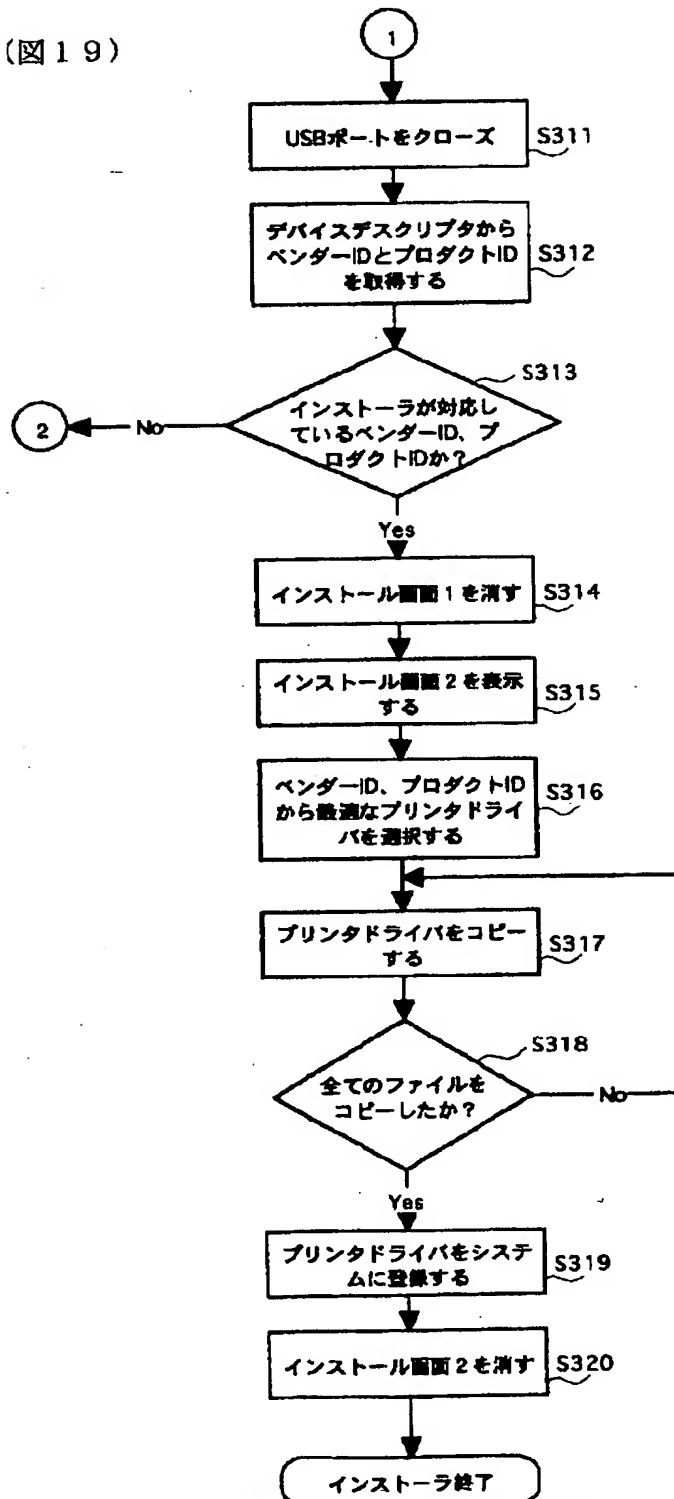


実施形態4のフローチャート前半

(図18)

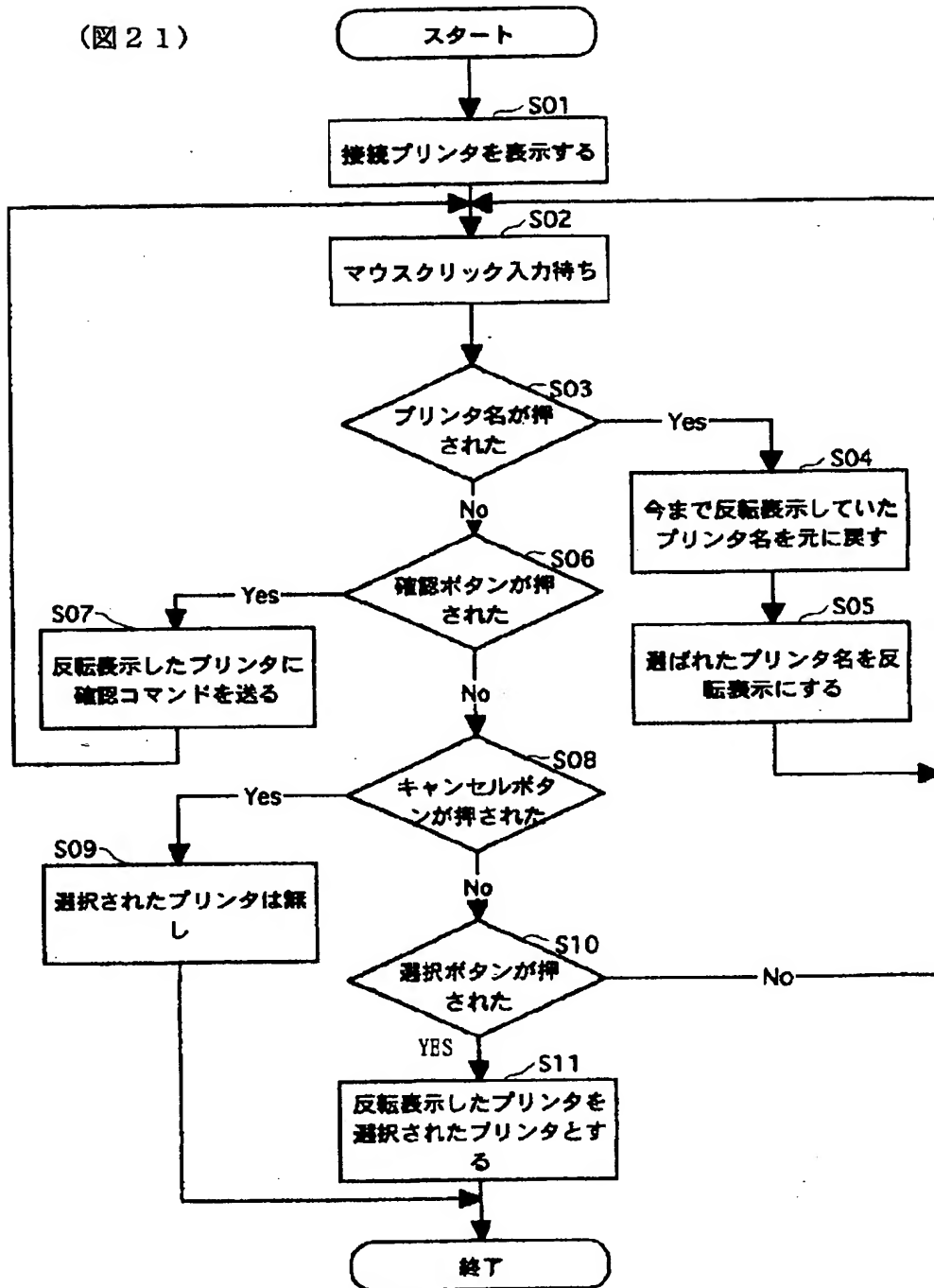
【図19】

(図19)



実施形態4のフローチャート後半

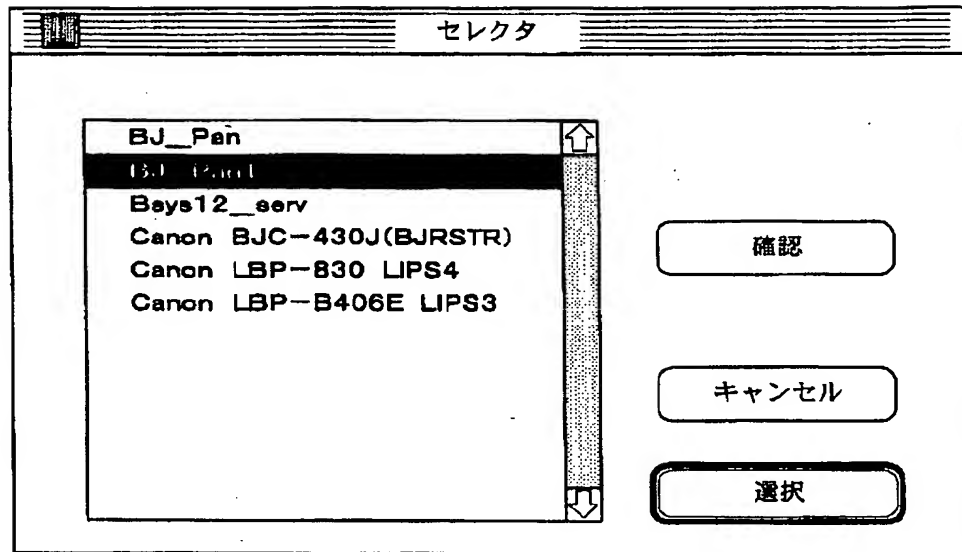
【図21】



実施形態5のフローチャート

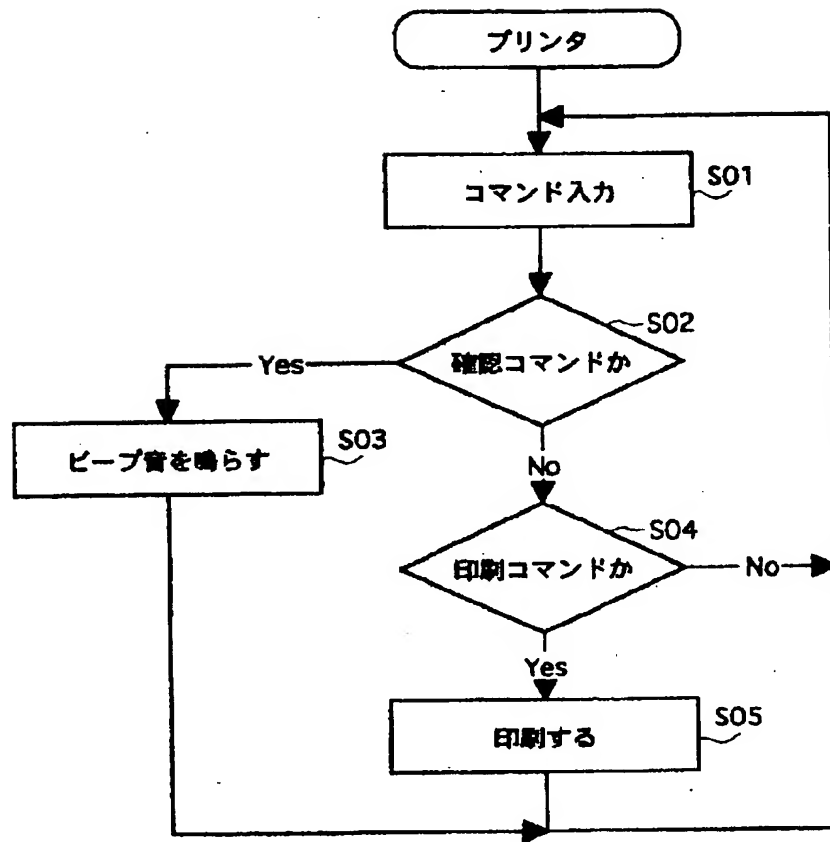
【図22】

(図22)



実施形態5のプリンタ選択ダイアログ

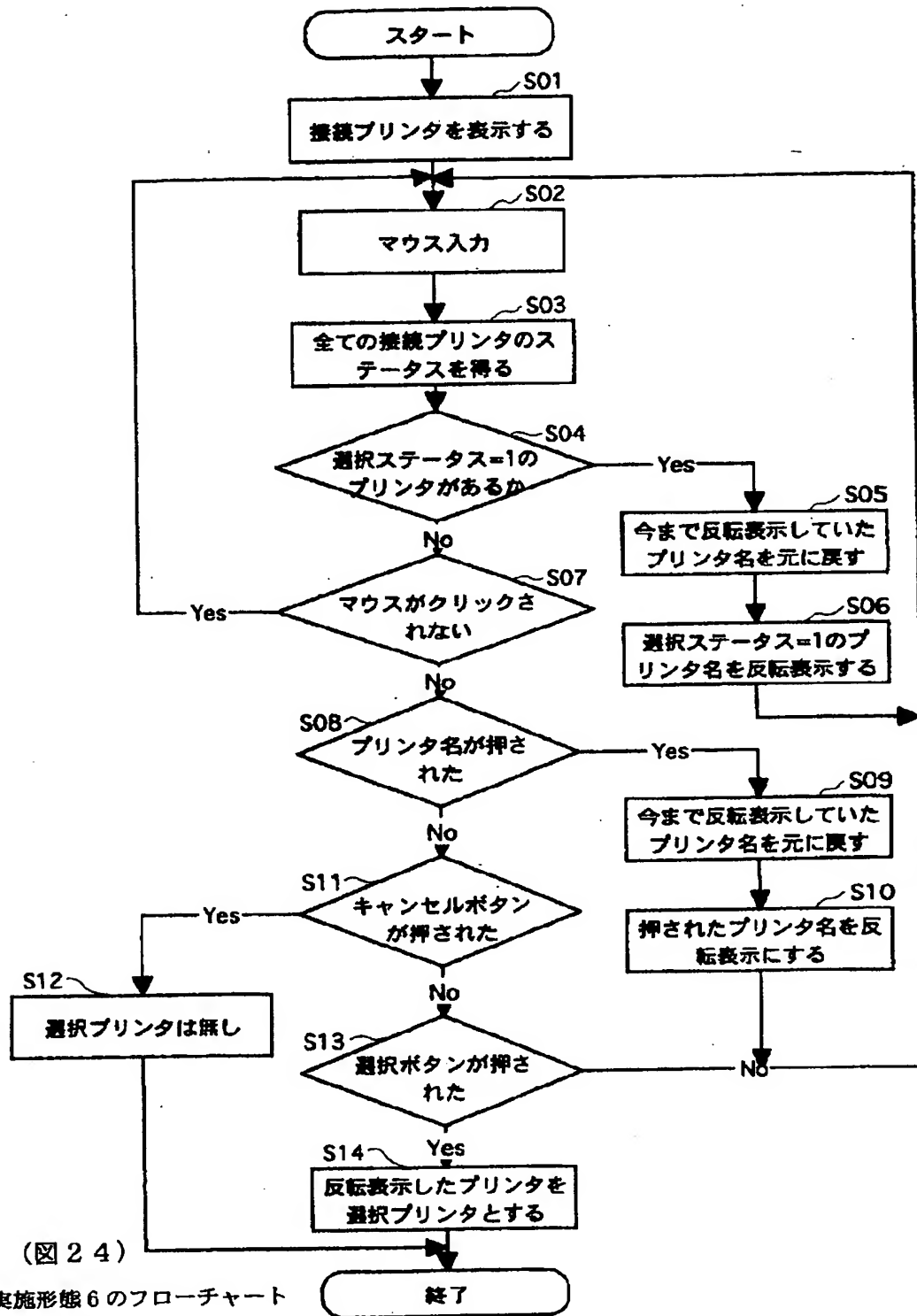
【図23】



プリンタのフローチャート

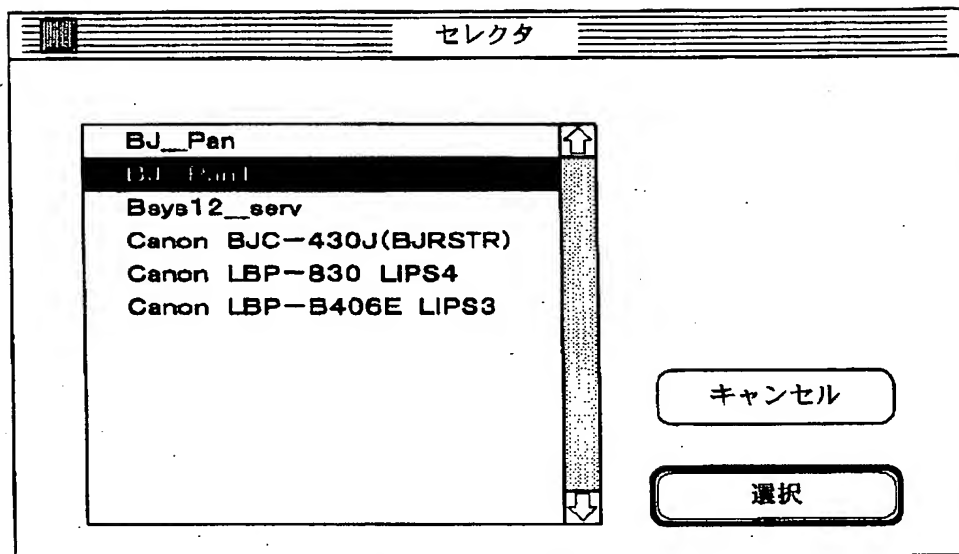
(図23)

【図 24】



【図25】

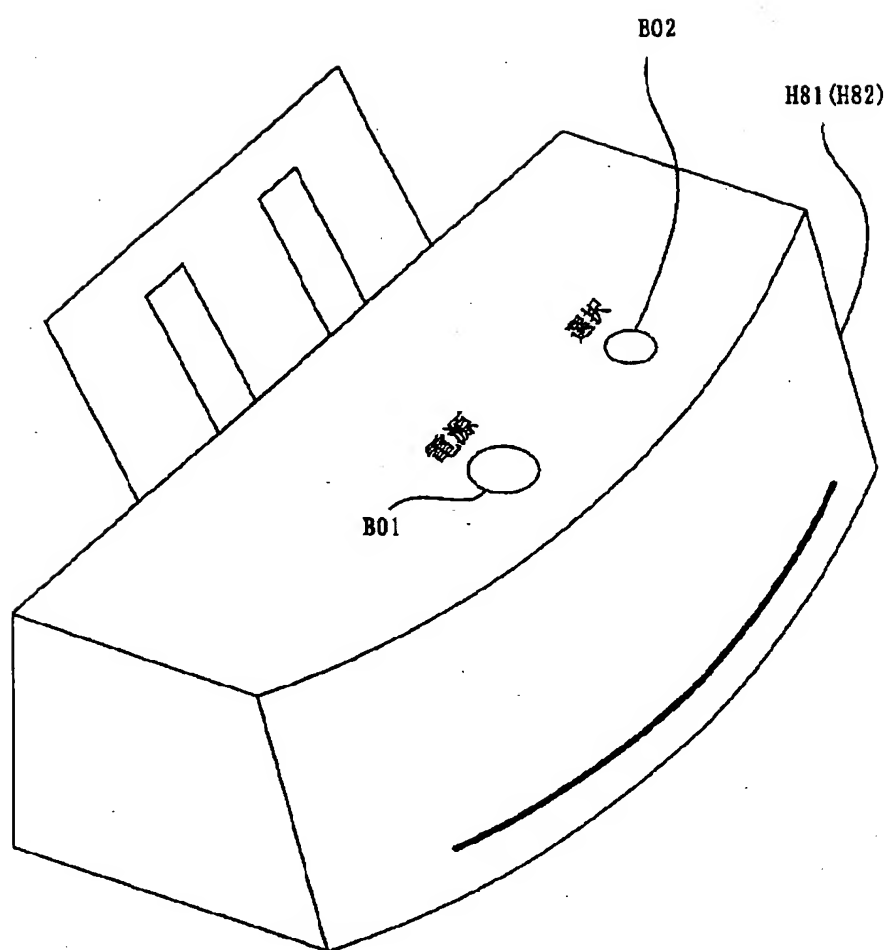
(図25)



実施形態6のプリンタ選択ダイアログ

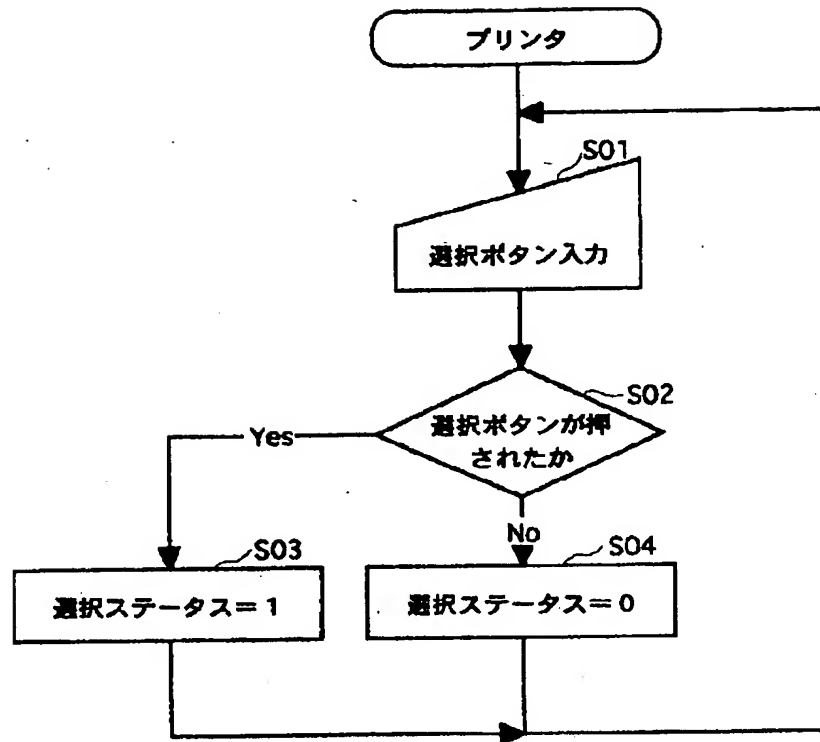
【図26】

(図26)



実施形態6の選択ボタン付きプリンタ

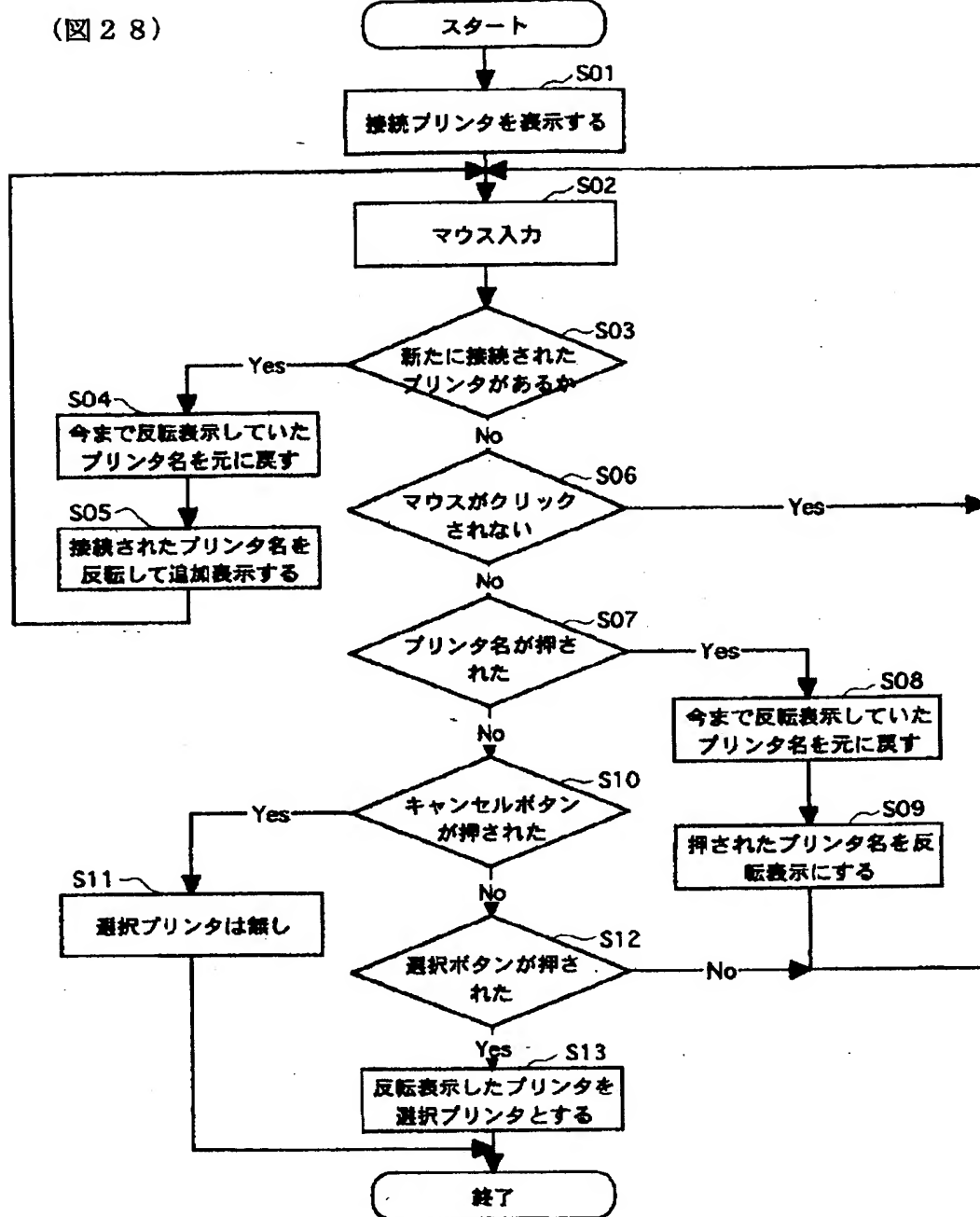
【図27】



選択ステータスのフローチャート

(図27)

【図 28】



実施形態 7 のフローチャート